Exponential Precision in A/D Conversion with an Imperfect Quantizer *

I. Daubechies, R. DeVore, C.S. Güntürk, and V. Vaishampayan

September 14, 2001

Abstract

We analyze mathematically the effect of quantization error in the circuit implementation of Analog to Digital (A/D) converters such as Pulse Code Modulation (PCM) and Sigma-Delta Modulation ($\Sigma\Delta$). We show that $\Sigma\Delta$ modulation, which is based on oversampling the signal, has a self correction for quantization error that is not inherited by PCM. This result may partially explain the success of such converters. Motivated by this example, we investigate whether it is possible to use redundancy to construct other encoders, with the same self correction property, but with higher order accuracy relative to bit rate. We introduce a class of encoders which exhibit exponential bit rate accuracy (in contrast to the polynomial rate of $\Sigma\Delta$) and still retain the self correction feature.

Keywords: A/D conversion, quantization, robustness, sigma-delta modulation, beta-expansions.

*This work has been supported by the National Science Foundation Grants DMS-9872890, DMS-9706753 and DMS-9729992, by the Office of Naval Research Contract N0014-91-J1343, and by Air Force for Scientific Research Contract F 49620-98-1-0044.

1 Introduction

A one-bit quantizer is a mapping Q of the real numbers into the discrete set $\{-1, 1\}$, defined by

$$Q(z) := \begin{cases} -1, & z \leq 0 \\ 1, & z > 0. \end{cases} \tag{1.1}$$

The hardware circuit implementation of such a device is never perfect: transition often happens at some point different from 0. This results in an erroneous quantizer $\tilde{Q}$, which instead computes

$$\tilde{Q}(z) := \begin{cases} -1, & z \leq \rho \\ 1, & z > \rho, \end{cases} \tag{1.2}$$

for a possibly unknown (small) value of ρ. One may further assume ρ to vary at each implementation of $\tilde{Q}$. We are interested in methods for converting an analog signal $x(t)$ defined on the real line $\mathbb{R}$ into a digital bitstream using such an imprecise one-bit quantizer.

Another essential ingredient of the methods we will be considering is the sampling operation, which maps a given signal $x(t)$ to a sequence of numbers $(x(n\tau))_{n\in\mathbb{Z}}$, where τ is the sampling interval. We assume that this operation is carried out precisely.

We shall work with bandlimited functions, i.e., functions whose Fourier transforms are compactly supported. Any bandlimited function can be recovered perfectly from its samples on a sufficiently close-spaced grid; this is known as the "sampling theorem". Let $\mathcal{S}(\Omega)$ denote the class of functions $x \in L^2(\mathbb{R})$ whose Fourier transforms are supported on $[-\Omega, \Omega]$. The Shannon-Whittaker formula gives a way to reconstruct a function $x \in \mathcal{S}(\pi)$ from its samples $(x(n))_{n\in\mathbb{Z}}$ taken on the integer grid:

$$x(t) = \sum_{n\in\mathbb{Z}} x(n)S(t-n), \tag{1.3}$$

where S is the sinc function

$$S(t) := \frac{\sin \pi t}{\pi t}. \tag{1.4}$$

The functions $S(\cdot - n)$, $n \in \mathbb{Z}$, form a complete orthonormal system for $\mathcal{S}(\pi)$. Clearly, the formula above can be extended through dilation to functions in $\mathcal{S}(\Omega)$ for arbitrary Ω.

In practice, one observes a function only on a finite portion $I = [a, b]$ of the real line $\mathbb{R}$. In addition, we shall consider functions of limited maximum amplitude; in other words, we consider the class $\mathcal{S}(\Omega, M, I)$ of all signals $x \in \mathcal{S}(\Omega)$ that take values in $(-M, M)$ when $t \in I$:

$$|x(t)| < M, \quad t \in I. \tag{1.5}$$

It will be sufficient in all of what follows to consider the case where $\Omega = \pi$ and $M = 1$. We denote $\mathcal{S}(\pi, 1, I)$ simply by $\mathcal{S}$.

Pulse Code Modulation (PCM) is maybe the simplest method for analog to digital conversion of functions in $\mathcal{S}$. Each sample $x(n) \in (-1, 1)$ in the expression (1.3) is simply replaced by a truncated version $\tilde{x}(n)$ of its binary expansion. (There is, however, a slight glitch to overcome due to the instability of the basis functions $S(\cdot - n)$, $n \in \mathbb{Z}$, which is reflected by the fact that

$$\sum_{n \in \mathbb{Z}} |S(t - n)| = \infty \tag{1.6}$$

whenever t is not an integer. This can be easily fixed by oversampling, as we shall see in Section 2.)

Let the real number $y \in (-1, 1)$ have the binary expansion

$$y = b_0 \sum_{i=1}^{\infty} b_i 2^{-i}, \tag{1.7}$$

with $b_0 = b_0(y) \in \{-1, 1\}$ and $b_i = b_i(y) \in \{0, 1\}$ for all $i \geq 1$. The sign bit b_0 is given by $b_0(y) = Q(y)$. The other bits can be computed using the one-bit quantizer described in (1.1) in the following algorithm known as Successive Approximation (SA). For each real number z, let $Q_1(z) := (Q(z-1)+1)/2$, i.e.,

$$Q_1(z) := \begin{cases} 0, & z \leq 1 \\ 1, & z > 1. \end{cases} \tag{1.8}$$

Let $u_1 := 2b_0 y = 2|y|$; the first bit b_1 is given by $b_1 := Q_1(u_1)$. Then the remaining bits are computed recursively as follows: if u_n and b_n have been defined, we let

$$u_{i+1} := 2(u_i - b_i) \tag{1.9}$$

and

$$b_{i+1} := Q_1(u_{i+1}). \tag{1.10}$$

Let us now consider what will happen if we make errors in the quantization. We suppose that at each quantization step, the circuit does not compute $Q(z)$ but rather

$\tilde{Q}(z)$, given by (1.2). This also leads to the definition

$$\tilde{Q}_1(z) := \begin{cases} 0, & z \le 1+\rho \\ 1, & z > 1+\rho, \end{cases} \tag{1.11}$$

where ρ may vary at each implementation of $\tilde{Q}_1$. We assume that $|\rho| \le \delta$ where $\delta > 0$ is fixed. The debilitating effect of the quantization error can already be seen in the sign bit. Assume for example that $\rho > 0$. If $y \in (0, \rho]$, then the sign bit of y will be incorrect. No matter how the remaining bits are assigned the resulting error $|y - \tilde{y}|$ is at least as large as $|y|$ which can be as large as δ. By taking a function $x(t) = yS(t-k)$, with S the sinc function (1.4), this translates into the same possible error for PCM in its circuit implementation. Note that this is not just an anomaly of only the sign bit. For $y \in (1/2, 1/2 + \rho/2)$, the sign bit $b_0(y)$ will be correct, but the bit $b_1(y)$ will be wrong and no matter how the other bits are assigned the resulting error $|y - \tilde{y}|$ will be at least as large as $|y - 1/2|$ which could be as large as $\delta/2$.

In this paper, we shall look at two other schemes which do not suffer from this effect. In these schemes, it will be possible to reconstruct the signals perfectly by taking more bits from their digital representations, even if the quantizer used to derive the digital representations is imperfect. The compensation will come, as we shall see, from the redundancy of the codewords produced by these algorithms. To give a more systematic treatment, we first give a short discussion of the information theoretical aspects of the problem in the next section.

2 Encoding-Decoding and Kolmogorov Entropy

Let $\mathcal{S}$ be the space of our signals, as defined in Section 1. By an encoder E for $\mathcal{S}$ we mean a mapping

$$E : \mathcal{S} \to \mathcal{B} \tag{2.1}$$

where the elements in $\mathcal{B}$ are finite bitstreams. The result of the encoding is to take the analog signal x to the digital domain. We also have a decoder D which maps bitstreams to signals

$$D : \mathcal{B} \to \tilde{\mathcal{S}} \tag{2.2}$$

where the class $\tilde{\mathcal{S}}$ is not necessarily the same as $\mathcal{S}$. Note that there may be many decoders D associated to a given E.

In general $DEx \neq x$. We can measure the distortion in the encoding/decoding by

$$\|x - DEx\| \tag{2.3}$$

where $\|\cdot\|$ is a norm defined on the signals of interest. We shall restrict our attention to the norm

$$\|x\|_{C(I)} := \sup_{t \in I} |x(t)|, \tag{2.4}$$

where $C(I)$ denotes the class of continuous functions on I. A similar analysis can be carried out for other norms such as the $L^2(I)$ norm.

One way of possibly assessing the performance of encoders is to measure the distortion of the encoding-decoding

$$d(\mathcal{S}; E, D) := \sup_{x \in \mathcal{S}} \|x - DEx\|_{C(I)}. \tag{2.5}$$

for the class $\mathcal{S}$. In order to have a fair competition between various encoding-decoding schemes we can look at the performance as a function of the bit budget. Given a distortion $\epsilon > 0$, we let $\mathcal{E} := \mathcal{E}(\mathcal{S}, \epsilon)$ denote the class of encoder-decoder pairs (E, D) for which $d(\mathcal{S}; E, D) \leq \epsilon$. A given encoder E corresponding to a pair in this class utilizes a bit budget

$$n(\mathcal{S}, \epsilon, E, I) := \sup_{x \in \mathcal{S}} \#(Ex), \tag{2.6}$$

where $\#(Ex)$ is the number of bits in the bitstream Ex. The smallest number of bits that can realize the distortion ϵ for $\mathcal{S}$ on I is given by

$$n(\mathcal{S}, \epsilon, I) := \min_{(E,D) \in \mathcal{E}(\mathcal{S}, \epsilon)} n(\mathcal{S}, \epsilon, E, I). \tag{2.7}$$

It is well known that $n(\mathcal{S}, \epsilon, I)$ is determined by the Kolmogorov entropy $\mathcal{H}_\epsilon(\mathcal{S}, C(I))$ of the class $\mathcal{S}$ as a subset of $C(I)$. Here $\mathcal{H}_\epsilon(\mathcal{S}, C(I)) := \log_2 N_\epsilon(\mathcal{S})$ where $N := N_\epsilon(\mathcal{S})$ is the smallest number of functions $f_1, \ldots, f_N \in C(I)$ such that each $f \in \mathcal{S}$ satisfies $\|f - f_i\|_{C(I)} \leq \epsilon$ for some $i = 1, 2, \ldots, N$. It is then easy to see that

$$n(\mathcal{S}, \epsilon, I) = \lceil \mathcal{H}_\epsilon(\mathcal{S}, C(I)) \rceil$$

with $\lceil y \rceil$ the smallest integer $\geq y$.

In many applications in signal processing, the interval I on which we wish to recover the signals $x(t)$ is varying and will be large relative to the Nyquist sample spacing. Consider, for example, audio signals where sampling at $44,000$ times per second corresponds to high quality audio and sampling at $4,000$ times per second corresponds to wireless voice applications. Recovering M minutes of such signals in these two cases means that the interval I corresponds to a number of $2,640,000M$ or $240,000M$ consecutive samples, respectively. So, the interval I, while finite, is typically very large. In this case, the average bit rate

$$\bar{n}(\mathcal{S},\epsilon) := \lim_{|I|\to\infty} \frac{n(\mathcal{S},\epsilon,I)}{|I|} \tag{2.8}$$

is a relevant measure of the encoding efficiency. There is a corresponding concept of average entropy

$$\bar{\mathcal{H}}_\epsilon(\mathcal{S}) := \lim_{|I|\to\infty} \frac{\mathcal{H}_\epsilon(S, C(I))}{|I|} \tag{2.9}$$

and we have

$$\bar{n}(\mathcal{S},\epsilon) = \bar{\mathcal{H}}_\epsilon(\mathcal{S}).$$

The average entropy of the class $\mathcal{S}$ can be derived from results of Kolmogorov and Tikhomirov [2] on the average entropy of classes of analytic functions. These results yield

$$\bar{n}(\mathcal{S},\epsilon) = (1+o(1)) \log_2 \frac{1}{\epsilon}, \quad 0 < \epsilon < 1. \tag{2.10}$$

One can derive this result through the practical encoding scheme PCM. We shall here give a formulation of the proof using methods that are similar to those used in [2], however, with more generality. The first issue to be taken care of is the instability of the expansion (1.3) as stated in (1.6). We can circumvent this problem easily in the following way. We choose a real number $\lambda > 1$ and sample x at the points n/λ, $n \in \mathbb{Z}$, thereby obtaining the sequence $x_n := x(n/\lambda)$, $n \in \mathbb{Z}$. When we reconstruct x from these sample values, we have more flexibility since the sample values are redundant. If $g := g_\lambda$ is any function whose Fourier transform $\hat{g}$ is 1 on $[-\pi,\pi]$, and vanishes outside of $[-\lambda\pi,\lambda\pi]$, then we have the recovery formula (see [1])

$$x(t) = \frac{1}{\lambda} \sum_{n\in\mathbb{Z}} x_n g(t - n/\lambda). \tag{2.11}$$

Let $G_\lambda(n) := \sup_{t\in[0,1/\lambda)} |g(t-n/\lambda)|$. We shall only need to consider g for which

$$\sum_{n\in\mathbb{Z}} G_\lambda(n) < \infty, \tag{2.12}$$

which can easily be achieved by choosing $\hat{g}$ to be sufficiently smooth.

Given an integer $m > 0$, we define the encoder E_m as follows. Given an interval $I = [a, b]$ in which we want to recover the signal, let $\bar{I} := [a - M, b + M]$ where $M = M(m, \lambda)$ is chosen so that

$$\sum_{|n| \geq \lfloor \lambda M \rfloor} G_\lambda(n) < 2^{-m}. \tag{2.13}$$

For $x \in \mathcal{S}$, we define $E_m(x)$ to be the bitstream consisting of the bits $b_i(x_n)$, $i = 0, \ldots, m$, $x_n \in \bar{I}$. Thus $\bar{x}_n := b_0(x_n) \sum_{i=1}^m b_i(x_n) 2^{-i}$ satisfies

$$|x_n - \bar{x}_n| \leq 2^{-m}. \tag{2.14}$$

¿From this bitstream, we can decode using the recovery

$$\bar{x}(t) := \frac{1}{\lambda} \sum_{n \in \lambda \bar{I}} \bar{x}_n g(t - n/\lambda). \tag{2.15}$$

We can easily bound the distortion for this encoding/decoding. We have

$$x = S_0 + S_1 \tag{2.16}$$

where $S_0 := \frac{1}{\lambda} \sum_{n \in \lambda \bar{I}} x_n g(t - n/\lambda)$ and $S_1 := \frac{1}{\lambda} \sum_{n \notin \lambda \bar{I}} x_n g(t - n/\lambda)$. Therefore,

$$|x(t) - \bar{x}(t)| \leq |S_1(t)| + |\bar{x}(t) - S_0(t)|. \tag{2.17}$$

We have $|g(t - n/\lambda)| \leq G_\lambda(n - \lfloor \lambda t \rfloor)$, so that for $t \in I$,

$$|S_1(t)| \leq \sum_{n \notin \lambda \bar{I}} |g(t - n/\lambda)| \leq \sum_{|k| \geq \lfloor \lambda M \rfloor} G_\lambda(k) \leq 2^{-m}. \tag{2.18}$$

In view of (2.14)

$$|\bar{x}(t) - S_0(t)| \leq 2^{-m} \sum_{n \in \lambda \bar{I}} |g(t - n/\lambda)| \leq C 2^{-m}, \tag{2.19}$$

where $C := C_\lambda$ denotes the sum of the series given by (2.12). Putting these two estimates into (2.17), we obtain

$$|x(t) - \bar{x}(t)| \leq (1 + C) 2^{-m}, \quad t \in I. \tag{2.20}$$

The number of bits used in this encoding is bounded by

$$m |\bar{I}| \lambda = m(2M + |I|)\lambda, \tag{2.21}$$

from which we derive

$$\bar{n}(\mathcal{S}, (1+C)2^{-m}) \leq \lim_{|I|\to\infty} |I|^{-1}(2M + \lambda|I|)m \leq \lambda m. \tag{2.22}$$

In other words,

$$\bar{n}(\mathcal{S}, \epsilon) \leq \lambda(\log_2 \frac{1}{\epsilon} + \log_2(1+C)). \tag{2.23}$$

Now, given any $\delta > 0$, we choose $\lambda > 1$ so that $\lambda - 1 \leq \delta/2$. Then for $\epsilon > 0$ sufficiently small, (2.23) gives

$$\bar{n}(\mathcal{S}, \epsilon) \leq (1+\delta)\log_2 \frac{1}{\epsilon} \tag{2.24}$$

We therefore obtain a proof of the upper bound of $\bar{n}(\mathcal{S}, \epsilon)$ stated in (2.10).

3 The Error Correction of Sigma-Delta Modulation

We have seen that for sufficiently large intervals I and a sufficiently large number of bits m per Nyquist interval, the accuracy $d(\mathcal{S}; E, D)$ of PCM, when implemented using a precise quantizer, is of order $O(2^{-m})$. However, when the quantizer is imperfect as defined in Section 1, the accuracy has no asymptotic decay as m is increased, as shown by our earlier examples. In this section, we shall look at another class of encoders, given by Sigma-Delta ($\Sigma\Delta$) Modulation, which behave differently when quantization error is present.

We describe here only the simplest case of first order $\Sigma\Delta$ modulation. We again choose $\lambda > 1$ (although now λ will be chosen to be large rather than close to one). We continue with the notation $x_n := x(n/\lambda)$ as above. Given (x_n), the algorithm creates a bitstream (b_n), $b_n \in \{-1, 1\}$, whose running sums are trying to match those of x_n. There is an auxiliary sequence u_n which tracks the error between these two running sums. We take $u_n = 0$ for $n < \lambda a$ where $I = [a, b]$, and define

$$u_{n+1} := u_n + x_n - b_n, \quad n/\lambda \in I, \tag{3.1}$$

where

$$b_n := Q(u_n) \tag{3.2}$$

and Q is the quantizer defined by (1.1). The $\Sigma\Delta$ encoder E_λ maps x to the bitstream (b_n). We see that E_λ appropriates one bit for each sample, which corresponds to λ bits per Nyquist interval, therefore resulting in $|I|\lambda$ bits for the whole signal x.

To decode the $\Sigma\Delta$ bitstream, we can use the recovery formula (2.15) with g as before and with each $\bar{x}_n$ replaced by b_n. This gives

$$\bar{x}(t) := D((b_n))(t) := \frac{1}{\lambda} \sum_{n \in \lambda \bar{I}} b_n g(t - n/\lambda). \tag{3.3}$$

One can easily show (see also [1]) that using appropriate g yields

$$|x(t) - \bar{x}(t)| \leq C_0 \lambda^{-1}, \quad t \in I, \tag{3.4}$$

with the constant C_0 depending only on the choice of g. Examples show that for general x, this bound cannot be improved, i.e.,

$$c_0 \lambda^{-1} \leq d(\mathcal{S}; E, D) \leq C_0 \lambda^{-1}. \tag{3.5}$$

Hence we see that when the average bit rate is λ, first order $\Sigma\Delta$ modulation results in $O(\lambda^{-1})$ accuracy, much worse than what would be achieved if PCM was used. However, the remarkable fact is that $\Sigma\Delta$ encoders are impervious to error in the circuit implementation of the quantization. Theorem 3.1, below, shows that given any $\delta > 0$, then an error of at most δ in each implementation of quantization in the $\Sigma\Delta$ encoder will not affect the distortion bound (3.4) save for the constant C.

Let us see how this works. Suppose that in place of the quantizer Q of (1.1), we use the imprecise quantizer $\tilde{Q}$ of (1.2), where ρ can vary at each occurence. We assume the uniform bound $|\rho| \leq \delta$. Using these quantizers will result in a different bitstream than would be produced by using Q. In place of the auxiliary sequence u_n of (3.1) which would be the result of exact quantization, we obtain the sequence $\tilde{u}_n$, which satisfies $\tilde{u}_n = 0$ for $n/\lambda < a$ and

$$\tilde{u}_{n+1} = \tilde{u}_n + x_n - \tilde{b}_n, \quad n \in \lambda I \tag{3.6}$$

where

$$\tilde{b}_n = \tilde{Q}(\tilde{u}_n). \tag{3.7}$$

We then have:

Theorem 3.1 *Suppose that $\Sigma\Delta$ modulation is implemented by using, at each occurence, one of the quantizers $\tilde{Q}$, with $|\rho| \leq \delta$, in place of Q. If the sequence $(\tilde{b}_n)$ is used in place of b_n in the decoder* (3.3)*, the result is a function $\tilde{x}$ which satisifies*

$$|x(t) - \tilde{x}(t)| \leq C\lambda^{-1}, \quad t \in I, \tag{3.8}$$

with $C = C_0(2+\delta)$ *and* C_0 *the constant in* (3.4) .

Proof: This theorem was proved in [1]. We do not repeat its simple proof in detail. The main idea which is to establish the following bound:

$$|\tilde{u}_n| \leq 2 + \delta, \tag{3.9}$$

which can be proved by induction on n. It is clear for $n < \lambda a$. Assume that (3.9) has been shown for $n \leq N$. If $\tilde{u}_N \leq \rho$, then $\tilde{b}_N = -1$ and from (3.6) we have

$$\tilde{u}_{N+1} = \tilde{u}_N + x_N - \tilde{b}_N. \tag{3.10}$$

Now, $x_N - \tilde{b}_N \in [0, 2]$ and hence $\tilde{u}_{N+1} \in [-2-\delta, 2+\delta]$. A similar argument applies if $\tilde{u}_N > \rho$ and therefore we have advanced the induction hypothesis and thus proved (3.9). The remainder of the proof uses summation by parts to obtain (3.8) (see [1]). □

The error correction capability in $\Sigma\Delta$ is related to the large amount of redundancy in the representation (2.11). The question arises whether one could utilize redundancy to build other encoders which have the best of both worlds: self correction for quantization error and exponential accuracy in terms of the bit rate. In the next section, we shall construct a class of encoders which have the flavor of PCM but rather than using the binary representation of a real number y (which is unique), they utilize representations with respect to a base $\beta \in (1, 2)$. Such beta-representations are not unique, even when β is kept fixed, and this fact is exploited to achieve the above mentioned properties.

4 Beta-Encoders with Error Correction

We shall now show that it is possible to obtain exponential bit rate performance while retaining quantization error correction by using what we shall call *beta-encoders*. The essential idea is to replace the binary representation of a real number y by a redundant representation.

Let $1 < \beta < 2$ and $\gamma := 1/\beta$. Then each $y \in [0, 1]$ has a representation

$$y = \sum_{i=1}^{\infty} b_i \gamma^i \tag{4.1}$$

with

$$b_i \in \{0, 1\}. \tag{4.2}$$

In fact there are many such representations. The main observation that we shall utilize below is that no matter what bits b_i, $i = 1, \dots, m$, have been assigned, then, as long as

$$y - \frac{\gamma^{m+1}}{1-\gamma} \le \sum_{i=1}^{m} b_i \gamma^i \le y, \tag{4.3}$$

there is a bit assignment $(b_k)_{k>m}$, which, when used with the previously assigned bits, will exactly recover y.

We shall use this observation in an analogous fashion to Successive Approximation to encode real numbers, with the added feature of quantization error correction. These encoders have a certain offset parameter μ whose purpose is to make sure that even when there is an imprecise implementation of the encoder, the bits assigned will satisfy (4.3); as shown below, introducing μ corresponds to carrying out the decision to set a bit to 1 only when the input is well past its minimum threshold. We let Q_1 be the quantizer of (1.8).

The beta-encoder with offset μ. *Let $\mu > 0$ and $1 < \beta < 2$. For $y \in [0, 1]$, we define $u_1 := \beta y$ and $b_1 := Q_1(u_1 - \mu)$. In general, if u_i and b_i have been defined, we let*

$$u_{i+1} := \beta(u_i - b_i), \quad b_{i+1} := Q_1(u_{i+1} - \mu). \tag{4.4}$$

It then follows that

$$y - \sum_{i=1}^{m} b_i \gamma^i = y - \sum_{i=1}^{m} \gamma^i (u_i - \gamma u_{i+1}) = y - \gamma u_1 + \gamma^{m+1} u_{m+1} \le \gamma^{m+1} \|u\|_{l^\infty}, \tag{4.5}$$

showing that we have exponential precision in our reconstruction, provided the $|u_i|$ are uniformly bounded. We shall see below that we do indeed have such a uniform bound. Let's analyze the error correcting abilities of these encoders when the quantization is imprecise. Suppose that in place of the quantizer Q_1, we use at each iteration in the beta-encoder the imprecise quantizer $\tilde{Q}_1$ defined by (1.11) where at each application the value of ρ may vary. We assume a uniform bound $|\rho| \le \delta$ for the quantizer errors. In place of the bits $b_i(y)$, we shall obtain inaccurate bits $\tilde{b}_i(y)$ which are defined recursively by $\tilde{u}_1 := \beta y$, $\tilde{b}_1 := \tilde{Q}_1(\tilde{u}_1 - \mu)$ and more generally,

$$\tilde{u}_{i+1} := \beta(\tilde{u}_i - \tilde{b}_i), \quad \tilde{b}_{i+1} := \tilde{Q}_1(\tilde{u}_{i+1} - \mu). \tag{4.6}$$

Theorem 4.1 *Let $\delta > 0$ and $y \in [0, 1)$. Suppose that in the beta-encoding of y, the quantizer $\tilde{Q}_1$ is used in place of Q_1 at each occurence, with the values of ρ possibly varying but always satisfying $|\rho| \leq \delta$. If $\mu \geq \delta$ and β satisfies*

$$1 < \beta \leq \frac{2 + \mu + \delta}{1 + \mu + \delta}, \tag{4.7}$$

then for each $m \geq 1$, $\tilde{y}_m := \sum_{k=1}^{m} \tilde{b}_k \gamma^k$ satisfies

$$|y - \tilde{y}_m| \leq C\gamma^m, \quad m = 1, 2, \ldots, \tag{4.8}$$

with $C = 1 + \mu + \delta$.

Proof: We first claim that

$$0 \leq \tilde{u}_n \leq \beta(1 + \mu + \delta), \quad n = 1, 2, \ldots. \tag{4.9}$$

This is proved by induction on n. For $n = 1$ it is true because

$$\tilde{u}_1 := \beta y \leq \beta.$$

Assume that (4.9) has been proved for $n = N$. If $\tilde{b}_N = 0$, then $\tilde{u}_N \leq 1 + \mu + \delta$ and hence

$$0 \leq \tilde{u}_{N+1} = \beta \tilde{u}_N \leq \beta(1 + \mu + \delta), \tag{4.10}$$

as desired. If $\tilde{b}_N = 1$, then $\tilde{u}_N > 1 + \rho + \mu \geq 1 - \delta + \mu \geq 1$. Also, in this case,

$$0 \leq \tilde{u}_{N+1} = \beta(\tilde{u}_N - 1) \leq \beta[\beta(1 + \mu + \delta) - 1] \leq \beta(2 + \mu + \delta - 1) = \beta(1 + \mu + \delta) \tag{4.11}$$

where we have used (4.7). This advances the induction hypothesis and proves (4.9). On the other hand,

$$y - \tilde{y}_m = y - \sum_{k=1}^{m} \tilde{b}_k \gamma^k = \gamma \tilde{u}_1 - \sum_{k=1}^{m} \gamma^k (\tilde{u}_k - \gamma \tilde{u}_{k+1}) = \gamma^{m+1} \tilde{u}_{m+1}, \tag{4.12}$$

which together with (4.9) gives (4.8). □

(Note that, in the special case $\delta = 0$, the bound (4.9) shows that the $|u_n|$ in (4.5) are uniformly bounded, as claimed above.)

For signal encoding, we utilize the beta encoder as in PCM. Namely, we take $\lambda > 1$ and let $x_n := x(n/\lambda)$ as before. We would like to avoid keeping sign bits of the x_n. We can do this by replacing x_n by $x'_n := (x_n + 1)/2$. For each x'_n we keep the first m bits $b_1(x'_n), \ldots, b_m(x'_n)$ of the beta encoder applied to x'_n. To decode, we use the beta-encoder bits to approximately recover x'_n by

$$\bar{x}'_n := \sum_{k=1}^{m} b_k(x_n)\gamma^k \tag{4.13}$$

and then approximately recover x_n by

$$\bar{x}_n := 2\bar{x}'_n - 1 \tag{4.14}$$

which satisfies

$$|x_n - \bar{x}_n| \leq C\gamma^m, \quad n \in \mathbb{Z}, \tag{4.15}$$

Given a signal x, an integer $m > 0$, and the interval I, we define $\bar{x}$ as in (2.15) except that we use the $\bar{x}_n$ of (4.14).

Theorem 4.2 *For any $x \in \mathcal{S}$, the beta-encoder/decoder with m bits per sample satisfies*

$$|x(t) - \bar{x}(t)| \leq C\gamma^m, \quad t \in I, \tag{4.16}$$

with C depending only on the reconstruction filter g. Moreover, if in place of the exact quantizer Q_1, we use, at each iteration, a quantizer $\tilde{Q}_1$ given by (1.11)*, with ρ satisfiying $|\rho| \leq \delta$, then we still obtain the error bound* (4.16) *with the constant C now depending also on δ.*

Proof: If we define S_0 as in (2.16), then Theorem 4.1 gives

$$|S_0(t) - \bar{x}(t)| \leq C\gamma^m, \quad t \in I.$$

If we couple this with (2.18), we arrive at (4.16). □

Acknowledgment. This work was the result of a collaboration started during the Summer of 1999. The non-AT&T authors would like to thank AT&T for its hospitality and partial support (RDV and CSG). We would also like to thank the many AT&T researchers, in particular Jont Allen, with whom we had extensive and enjoyable discussions.

References

[1] I. Daubechies and R. DeVore, Reconstructing a bandlimited function from very coarsely quantized data: A family of stable sigma-delta modulators of arbitrary order, submitted.

[2] A. N. Kolmogorov and V. M. Tikhomirov, ϵ-entropy and ϵ-capacity of sets in function spaces. *Uspehi*, 14(86):3-86, 1959 (also: AMS Translations, Series 2, Vol. 17, 1961, pp. 277-364).

Ingrid Daubechies
Department of Mathematics and Program in
Applied and Computational Mathematics
Princeton University
Fine Hall, Washington Road
Princeton, NJ 08544.
email: `ingrid@math.princeton.edu`

Ronald DeVore
Department of Mathematics
University of South Carolina
Columbia, SC 29208.
email: `devore@math.sc.edu`

C. Sinan Güntürk
School of Mathematics
Institute for Advanced Study
Einstein Dr.
Princeton, NJ 08540.
email: `gunturk@math.ias.edu`

Vinay Vaishampayan
Information Sciences Research
AT&T Labs
180 Park Avenue
Florham Park, NJ 07932.
email: `vinay@research.att.com`