Sigma-Delta Quantization and Sturmian Words

by

ULAŞ AYAZ

B.Sc., Boğaziçi University, 2007

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE

in

The Faculty of Graduate Studies

(Mathematics)

THE UNIVERSITY OF BRITISH COLUMBIA

(Vancouver)

October 2009

Abstract

In this thesis, our main focus is Sigma-Delta quantization schemes. These are commonly used in state-of-art Analog-to-digital conversion technology. Their main advantage is the ease of implementation and more importantly their insensitivity to certain circuit imperfections. When we compare sigma-delta scheme with pulse-code modulation (PCM), sigma-delta is inferior in terms of rate distortion because an N-bit kth order sigma-delta quantizer produces an approximation with the error of order $O(N^{-k})$ whereas the corresponding N-bit PCM scheme has accuracy of $O(2^{-N})$. However, this is a raw estimate of the actual rate-distortion characteristic of sigma-delta as one can further compress the bitstreams obtained via sigma-delta quantization. Even though this observation was made earlier in [10] under certain assumptions, to our knowledge, it was not investigated fully. In this thesis, such an investigation is made for first-order sigma-delta quantizers by using some results from symbolic dynamics literature on "Sturmian words". Surprisingly, it turns out that the approximation error is a function of the "actual bit-rate", i.e., the bit-rate after compressing an N-bit first-order sigma-delta encoding.

In addition, in this thesis, we will introduce a new setup for sampling a bandlimited function and then quantizing these samples via first-order sigma-delta scheme. This simple but surprisingly efficient technique will allow us to get a better bound for the approximation rate of sigma-delta schemes and it will allow us to apply the derived results for compression of the bitstreams.

Table of Contents

List of Figures

Acknowledgements

I would like to thank my advisor Özgür Yılmaz for his help and support during my two years at UBC. He has been very patient and understanding with me as well as a very good mentor for a young mathematician at the beginning of his path. I also thank him for sharing his ideas generously with me and advising me during the whole period of writing this thesis.

Chapter 1

Introduction

In signal processing, one of the main issues is encoding an analog object with finitely many number of bits in an efficient and robust way. Typically, signals of interests, i.e., audio, video signals and natural images, take their values in the continuum. These signals need to be represented by discrete-valued finite sequences in order to digitally process and transmit this data for further applications. This is called as Analog-to-digital (A/D) conversion (see [3], [4], [5], [6]). A/D conversion process consists of two main steps: *sampling* and *quantization*. *Sampling* is mainly collecting real-valued sample values of a function x on a sufficiently dense grid in the domain of x, i.e., the analog signal which we want to digitally process. In the A/D conversion setting, typically, x belongs to a linear space X and enjoys a decomposition

$$x = \sum_{j \in \Lambda} x_j \psi_j$$

where $\{\psi_j\}_{j \in \Lambda}$ is a *basis* or a *frame* for X – sampling can be seen as obtaining the values of x_j in this decomposition. The goal in *quantization* is to replace the coefficients x_j, which are in general real numbers, with some $q_j \in \mathcal{A}$, where $\mathcal{A}$ is a a fixed finite set, such that $\|x - \sum q_j \psi_j\|$ is small for some approximation norm of interest.

To make our discussion more concrete, we now consider a common example: Let $X = B_\Omega$ where B_Ω is the space of bandlimited functions, i.e., $B_\Omega = \{x \in L^2 : \text{supp } \hat{x} \subseteq [-\Omega, \Omega]\}$. Here, $\hat{x}$ denotes the Fourier transform of x. In this case, if $x \in B_\Omega$ is sampled on a regular grid that is sufficiently dense, then x can be perfectly reconstructed from its sample values. In particular, x is completely characterized by it samples $\{x(n/2\lambda\Omega) : \ n \in \mathbb{Z}\}$ for any $\lambda \geq 1$. This is known as the *classical sampling theorem*. The quantity

$T = 1/2\lambda\Omega$ is called the *sampling rate* and λ is called the *oversampling ratio*. When $\lambda = 1$, the corresponding sampling rate is called the *Nyquist sampling rate*. We will visit bandlimited functions setting in Chapter 5 again.

In this thesis, our main emphasis is on quantizing real numbers (this can be thought of obtaining alternative "binary" representations for real numbers). In particular, we will fix $X = [0, 1]$ and investigate various representations of $x \in [0, 1]$ by bitstreams of finite length, say N. The two important criteria in evaluation of different representations are: i) *approximation rate* and ii) *robustness* of the representation. The approximation rate refers to the rate with which the associated approximation error decays as N increases. On the other hand, we say that a representation is robust if we can obtain it with imperfect arithmetic and still get a decent approximation error that goes to 0 as N goes to infinity. (See Section 2.1 for a more specific definition of robustness).

We will start with pulse-code-modulation (PCM) methods which are based on computing truncated binary expansions of x. These are the most intuitive quantizers and in fact they provide optimally accurate approximations, see [4]. On the other hand, it is well known that these schemes are not robust. This is the main motivation for considering alternative representations, and in particular Sigma-Delta ($\Sigma\Delta$) schemes. These schemes were first devised in 1960s, [14]. There has been active research on these schemes since then and they have been popular in A/D conversion of bandlimited functions. $\Sigma\Delta$ schemes produce approximations the accuracy of which behaves like an inverse polynomial in the number of bits used, and so they are significantly inferior when compared with PCM. However, $\Sigma\Delta$ schemes are robust with respect implementation imperfections, [3, 4, 6, 9, 19].

In this thesis, we investigate the extent to which representations of real numbers on $[0, 1]$ via first-order $\Sigma\Delta$ schemes can be compressed. In particular, it is known that not all 2^N distinct N-bit *words* can be produced by running a first-order $\Sigma\Delta$ scheme with some input $x \in [0, 1]$. In fact, it is shown in [10] and in [12] that for every fixed initial condition in $[0, 1]$ the number of distinct bit-streams of length N produced by $\Sigma\Delta$ is $O(N^2)$.

Using some results from the literature on *Sturmian words*, e.g. [16] and

[1], we will show that the standard first-order $\Sigma\Delta$ scheme (implemented perfectly) produces only $O(N^3)$ different N-bit words with any input x and any initial condition. Furthermore, we show that the set of all codewords (which has cardinality $O(N^3)$ remains unchanged if the scheme is implemented with some unknown (but fixed) quantization offset error. Consequently, the N-bit words obtained by such a scheme can be represented using $R = 3\log_2 N$ bits. It follows that after compression, the approximation rate of a first-order $\Sigma\Delta$ scheme becomes $O(2^{-\frac{R}{3}})$. Note that in this approach we see compression as a "post-processing" stage that is performed in the digital domain. In other words, such a compression stage does not compromise robustness properties of the first-order $\Sigma\Delta$ schemes. On the other hand, the above reasoning is based on counting all codewords that can be obtained using a first-order $\Sigma\Delta$ scheme. This number, unfortunately, is affected if the scheme is implemented with "random" offset errors. In Section 4.4, we provide sufficient conditions under which the scheme, even when it is implemented with random offset errors, generates the same $O(N^3)$ codewords. Consequently, if these conditions are satisfied, the approximation rate of an imperfectly-implemented first-order $\Sigma\Delta$ scheme remains to be $O(2^{-\frac{R}{3}})$.

The organization of this thesis is as follows. We devote Chapter 2 to the introduction of basic notions and setup for the general A/D conversion problem. Some important results regarding PCM schemes (e.g., they achieve the Kolmogorov ϵ-entropy (see [18]) when encoding restrictions of bandlimited functions to intervals, [3], [4]), important drawbacks of PCM, and basic definitions of first and higher-order $\Sigma\Delta$ schemes are also given in this chapter. Chapter 3 is on Sturmian words. In particular, we highlight some results in this theory and investigate their implications for counting codewords associated with a first-order $\Sigma\Delta$ quantizer. In Chapter 4, we analyze the affects of various offset error types on the number of distinct codewords that can be generated by a first-order $\Sigma\Delta$ quantizer. We state the main theorems of this thesis in this chapter. Finally, Chapter 5 investigates the implications of our earlier results in the bandlimited settings.

Chapter 2

Encoding and Robustness

2.1 A/D conversion setting

We start with setting the notation and describing the problem (see also [5]). Suppose that X is a compact metric space equipped with the metric induced by some norm $\|\cdot\|$. Our goal is to represent $x \in X$ by finitely many bits. Equivalently, we wish to find a map $\psi : X \to \{0,1\}^{\mathbb{N}}$ that is invertible. Given such a map, we define the associated *N-bit encoder* as:

$$E_N : x \to (\psi(x)_1, \psi(x)_2, \ldots, \psi(x)_N)$$

A map D_N is called a *decoder* if it maps the range of E_N to a subset of X. It can be observed that if X is a infinite set, E_N can never be one-to-one which means that A/D conversion has to be lossy. We define the *reconstruction error*, α for a given encoder, E_N, as follows:

$$\alpha(E_N) = \inf_{D_N} \sup_{x \in X} \|x - D_N(E_N(x))\|$$

Clearly, if ψ is invertible, $\alpha(E_N) \to 0$ as $N \to \infty$. One of the important qualities of an encoder is the rate at which $\alpha(E_N) \to 0$ as $N \to \infty$. This is called *approximation rate*. The limit of this rate is determined by the space X, more precisely by the Kolmogorov ϵ-entropy of X. This is defined to be the base-2 logarithm of the smallest number k such that there exists an ϵ-net for X of cardinality k (see [18]).

Another important criterion for an A/D conversion scheme is its robustness. Due to imperfections of the circuits that implement the encoding algorithm, while passing from an analog value to a discrete range, some

small perturbations are introduced. In other words, the quantization formula of the encoder is not fully known. This motivates the following formal definition of robustness.

Definition 2.1.1. (*Robustness*) Suppose that we have an encoder $\{E_N^\delta\}$ where δ is some parameter (e.g., quantizer threshold values). We say that $\{E_N^\delta\}$ is robust with respect to δ if there exists a decoder D_N for $E_N^{\delta_0}$ and $\epsilon > 0$ such that $\|x - D_N(E_N^\delta(x))\| \to 0$ as $N \to \infty$ whenever $d(\delta, \delta_0) \leq \epsilon$ (here d is an appropriate metric on the parameter space).

2.2 PCM encoders

We fix $X = [0, 1)$. Suppose that we want to encode $x \in X$ with N bits. A natural way to do that is finding the binary expansion of x and using the N-bit truncated approximation. Consider the dynamical system defined by

$$T : u \to \langle 2u \rangle$$

where $\langle 2u \rangle$ denotes the fractional part of $2u$, i.e., $\langle 2u \rangle = 2u - \lfloor 2u \rfloor$. Next, put

$$b_i := \begin{cases} 1 & \text{if } T^i(x) \in [0, \frac{1}{2}) \\ 0 & \text{if } T^i(x) \in [\frac{1}{2}, 1) \end{cases}, \ i = 1, 2, \ldots \tag{2.1}$$

The PCM encoder E_N^{PCM} and an associated decoder D_N^{PCM} can be defined as

$$E_N^{PCM}(x) = (b_1, b_2, \ldots, b_N)$$

and

$$D_N^{PCM}(\{b_i\}_{i=1}^N) = \sum_{i=1}^N b_i 2^{-i} + 2^{-N-1}.$$

It can then be proved that for all $x \in [0, 1)$

$$|x - D_N^{PCM}(E_N^{PCM}(x))| \leq 2^{-N-1}$$

and consequently,

$$\alpha(E_N^{PCM}) = O(2^{-N}). \tag{2.2}$$

PCM achieves Kolmogorov ϵ-entropy: Given a bit budget of N, the approximation rate (2.2) is the fastest rate that can be achieved by an encoder, i.e., PCM encoders achieve the Kolmogorov ϵ-entropy for $X = [0,1]$. The notion of Kolmogorov entropy [18] can be explained roughly by asking the following question: Given N bits, what is the smallest ϵ value such that we can have an ϵ-net for $X = [0,1]$ of cardinality 2^N? Here 2^N is the number of points we can distribute in this interval with N bits. If we distribute these points with equal spaces in the $[0,1]$ interval, then the distance between them, i.e., ϵ value needed, will be order of 2^{-N} which is the approximation rate achieved by the PCM encoders (for a discussion in a more general setting see [4]).

Robustness of PCM: PCM is optimal in approximation rate and remains the standard encoding scheme for audio signals but it is not that as popular as its optimality might suggest. This is mainly because PCM suffers from various implementation difficulties and imperfections. Note that x is an analog quantity, so $T^i(x)$ must be implemented on analog hardware. Realization of a binary expansion on a circuit, i.e., dividing the interval $[0,1]$ into 2^N equal bins, is challenging. Particularly, it can be very costly to implement a device which generates b_i sequence of the PCM encoder, especially in the high-accuracy setting, i.e., when N is large. (See [3–6, 9] for further discussion). Next we will see why PCM behaves poorly under certain circuit imperfections.

Proposition 2.2.1. *Define $\{E_N^{PCM,\delta}\}$ by replacing T in (2.1) with T^δ : $u \to 2u - \lfloor 2u + \delta \rfloor$ and note that $\{E_N^{PCM,0}\}$ coincides with the classical PCM encoder. Then*

$$\sup_{x\in[0,1)} |x - D_N^{PCM}(E_N^{PCM,\delta}(x))| \geq \delta/2.$$

Consequently, $E_N^{PCM,\delta}$ is not robust with respect to δ.

Proof. Suppose: $T^\delta : u \to 2u - \lfloor 2u + \delta \rfloor$. Then,

$$x = \frac{1}{2} - \frac{\delta}{2} \Rightarrow b_1 = 1, b_2 = 0, \ldots.$$

So, $|x - \sum_{i=1}^{N} b_i 2^{-i}| = \frac{\delta}{2}$ for all N. □

Thus, we cannot correct the error by choosing N larger.

2.3 Another family of encoders: ΣΔ schemes

Despite the superior performance of PCM quantizers in terms of approximation rate, their lack of robustness is a reason for investigation of different, more robust quantization methods. ΣΔ quantizers have been introduced in the 60s and found wide use in electrical engineering (see [13], [15] and [17] for a further reading). To quantize $x \in X$, one first finds an expansion of x in the form of $x = \sum x_n e_n$ where $\{e_n\}$ is a basis or a frame for X. Then, ΣΔ quantizers take the coefficients x_n as input and generate their output recursively. For example, one-bit ΣΔ quantizers replace each coefficient x_n with either -1 or 1. Although multi-bit ΣΔ quantizers are getting popular in applications, in this thesis, we will restrict our attention to one-bit quantizers (see [2], [7] and [14] for a general idea of one-bit quantizers). Let us start by giving the most general setup for a kth order ΣΔ quantizer. Define

$$\Delta_n^k(u) := \sum_{i=0}^{k} (-1)^l \binom{k}{l} u_{n-1}.$$

Note that $\Delta_n^0(u) = u_n$ and $\Delta_n^1(u) = u_n - u_{n-1}$. A kth order ΣΔ quantization scheme is defined by the following system of difference equations:

$$\begin{aligned} \Delta_n^k &= x_n - q_n \\ q_n &= \text{sign}(F(\Delta_n^0(u), \ldots, \Delta_n^{k-1}(u), x_n)) \end{aligned} \tag{2.3}$$

where F is an arbitrary function on $\mathbb{R}^{k+1}$ chosen so that u_n stays bounded. It is well known that the approximation rate for a stable kth-order ΣΔ quan-

tizer is order of $O(N^{-k})$, where N is the bit budget allocated for the quantization. This estimate falls short in comparison with PCM which achieves an exponential decay. There is a vast research area on deriving better bounds on reconstruction error, i.e., decaying faster with respect to N ([3], [9]). One other challenge with ΣΔ quantizers of order $k > 1$, is stability. A ΣΔ quantizer which has uniform bounds for state variables u_n is called to be *stable* and choosing an appropriate F is required to obtain a stable scheme. In [3] and [9], families of stable high-order ΣΔ quantizers are introduced.

2.3.1 First-order ΣΔ quantizers

Throughout this paper we will consider and analyze only first-order ΣΔ quantizers with constant input. An ideal first-order quantizer is defined via the recursion

$$\begin{aligned} &u_0 \in [0,1) \text{ (initial condition);} \\ &q_n = Q(u_{n-1} + x) \\ &u_n = u_{n-1} + x - q_n = \langle u_{n-1} + x \rangle, \ n = 1, 2, \dots \end{aligned} \tag{2.4}$$

where the *quantizing map* Q is given by

$$Q(u) = \begin{cases} 0 & u < 1 \\ 1 & u \geq 1 \end{cases} \tag{2.5}$$

In (2.4), the *input* x is in $[0,1)$, and $\langle r \rangle$ denotes the fractional part of r. Since $u_{n-1} + x$ remains in $[0,2)$ for all n, we replaced map Q by *floor function* in (2.4).

Stability and robustness: It is not difficult to observe that $u_n \in [0,1)$ for all n, so ensuring stability for an ideal first-order ΣΔ quantizer is not as challenging as it is for higher order schemes. In addition, a first-order ΣΔ quantizer is known to be robust with respect to circuit imperfections. In comparison to PCM, there are several reasons why ΣΔ schemes are more reliable when quantizer imperfections are considered. In the PCM case, bi-

nary expansions real numbers are essentially unique. So if any of the bits in the representation is erroneous, this error cannot be fixed by modifying the rest of the binary word. In contrast, in the case of ΣΔ quantization, every real number can produce many different codewords with similar approximation properties (mainly depending on the initial state of the circuit, i.e., u_0, as well as the quantizing map that is used -note that any map Q can be used in (2.4) as long as the resulting u_n are guaranteed to remain bounded). Because of this multitude of "good" codewords, errors that are introduced due to circuit imperfections can be implicitly corrected later in the quantization process.

In ΣΔ literature, it is common to work with schemes where the quantizer alphabet is $\{\pm 1\}$ rather than $\{0, 1\}$, e.g., [19, Equation (10)]:

$$\begin{aligned} &v_0 \in [-1, 1) \text{ (initial condition)} \\ &v_n = v_{n-1} + x - h_n \\ &h_n = \text{sign}(u_{n-1} + x) \end{aligned} \tag{2.6}$$

which is a special case of the higher order scheme given in (2.3).

Here h_n's are quantization outputs and v is an internal state variable with $v_0 \in [-1, 1)$ and it can be shown that if $x \in [-1, 1)$, then $v_n \in [-1, 1)$ for all n. For the sake of completeness, we now show that the former system (2.4) is indeed equivalent to the latter one, (2.6).

Proposition 2.3.1. *Let $x \in [-1, 1)$. Suppose that h_n are obtained via (2.6) with input x and initial condition $v_0 \in [-1, 1)$, and q_n are obtained by running (2.4) with input $\frac{x+1}{2}$ and initial condition $u_0 = \frac{v_0+1}{2}$. Then $q_n = 1$ whenever $h_n = 1$ and $q_n = 0$ whenever $h_n = -1$.*

Proof. We can re-write (2.4) as:

$$v_n + 1 = v_{n-1} + 1 + x + 1 - (h_n + 1) \tag{2.7}$$

Define $\overline{v}_n := v_n + 1$, $\overline{x} := x + 1$ and $\overline{h}_n := h_n + 1$. Then $\overline{v}_n,\ \overline{x} \in [0, 2)$. Then

we have:

$$\begin{aligned}\overline{v}_n &= \overline{v}_{n-1} + \overline{x} - \overline{h}_n \\ \overline{h}_n &= \text{sign}(\overline{v}_n - 1 + \overline{x} - 1) + 1 \\ &= 2\lfloor \frac{1}{2}(\overline{v}_{n-1} + \overline{x}) \rfloor \end{aligned} \tag{2.8}$$

and $\overline{x}, \overline{v}_n \in [0, 2)$. Now define $u_n := \frac{\overline{v}_n}{2}, \tilde{x} := \frac{\overline{x}}{2}$ and $q_n := \frac{\overline{h}_n}{2}$. So arranging (2.8) again, we have:

$$\begin{aligned} u_n &= u_{n-1} + \tilde{x} - q_n \\ q_n &= \lfloor u_{n-1} + \tilde{x} \rfloor \end{aligned} \tag{2.9}$$

where $\tilde{x}, u_n \in [0, 1)$ and $q_n = \frac{h_n + 1}{2}$. As one can observe the scheme given by (2.9) is identical to the original scheme (2.4). And there is a bijective mapping

$$\begin{aligned} X : [-1, 1) &\to [0, 1) \\ x &\to \frac{x+1}{2} = \tilde{x} \end{aligned}$$

for the inputs of two systems. So the proof is complete. □

Approximation rate with a simple decoder: In order to find the approximation rate we assume that the data is constructed by following decoder:

$$D_N^{\Sigma\Delta}(\{q_i\}_{i=1}^N) := \frac{1}{N} \sum_{i=1}^{N} q_i$$

then,

$$\left.\begin{aligned} u_1 - u_0 &= x - q_1 \\ u_2 - u_1 &= x - q_2 \\ &\vdots \\ u_N - u_{N-1} &= x - q_N \end{aligned}\right\} \; u_N - u_0 = Nx - \sum_{i=1}^{N} q_i \tag{2.10}$$

$$\Rightarrow x - \frac{1}{N}\sum_{i=1}^{N} q_i = \frac{1}{N}(u_N - u_0)$$

Note that $u_N \in [0,1)$ and $u_0 \in [0,1) \Rightarrow \; -1 < u_N - u_0 < 1$.

$$\left| x - \frac{1}{N}\sum_{i=1}^{N} q_i \right| \leq \frac{1}{N} \tag{2.11}$$

Improved approximation rates: One can improve the error bound (2.11) by using arguments from the theory of uniform distribution (along with a slightly more sophisticated reconstruction kernel). Güntürk in [8] uses the notion of *discrepancy* and some results on exponential sums to prove the following bound for a first-order ΣΔ quantizer:

$$\left| x - \sum_{i=1}^{N} h_i q_i \right| \leq C_x N^{-2} \log^{2+\epsilon} N$$

where $\{h_i\}_{i=1}^{N}$ is the triangular filter with $\sum_{i=1}^{N} h_i = 1$ and $\epsilon > 0$ is arbitrary.

Comparison of PCM and ΣΔ: So far we have two main results, (2.2) and (2.11) for the reconstruction error corresponding to PCM and first-order ΣΔ respectively. Clearly, the exponential accuracy obtained via PCM is significantly superior to the inverse polynomial accuracy of ΣΔ. In Chapter 3, we will derive some very useful results from the symbolic dynamics literature which will bound the actual number of codewords that can be generated by a first-order ΣΔ quantizer. This will allow us to further compress bitstreams coming out of a ΣΔ quantizer and at the end of the chapter we will give an alternative comparison of the approximation rates of PCM and ΣΔ

schemes after allowing a "post-compression" stage in the case of ΣΔ. This alternative angle illustrates that ΣΔ schemes have "exponential accuracy" after such a compression.

Chapter 3

$\Sigma\Delta$ Quantization and Sturmian Words

Recall that the recursion given in (2.4) describes an ideal first-order $\Sigma\Delta$ quantizer. As explained before, in order to make a fair comparison between $\Sigma\Delta$ quantization and PCM, we need to count the number of distinct codewords that can be generated by $\Sigma\Delta$ quantizer after N steps. In this chapter, we will show that using results on Sturmian words, we can achieve this task.

3.1 Sturmian words

Here we will rely heavily on results on Sturmian words by Mignosi [16]. Below, we will adopt the notation of [16] whenever possible.

Definition 3.1.1. [16, p.71] Let x and u_0 be two real numbers with $x \in [0,1)$. Consider the sequence $\langle (n-1)x + u_0 \rangle, n \in \mathbb{N}, n > 0$, and define the infinite word $f_{x,u_0} = b_1(x,u_0)b_2(x,u_0)b_3(x,u_0)\dots$ by the rule:

$$b_n(x,u_0) = \begin{cases} 0 & \text{if } \langle (n-1)x + u_0 \rangle \in [0, 1-x), \\ 1 & \text{if } \langle (n-1)x + u_0 \rangle \in [1-x, 1), \end{cases} \tag{3.1}$$

where for any real number r, $\langle r \rangle$ is the fractional part of r. We will write b_n instead of $b_n(x,u_0)$ whenever there is no possibility of confusion.

Proposition 3.1.2. *The first N elements generated by the equation* (3.1) *with input x and inital value u_0, say $\{b_i\}_{i=1}^N$, is identical to the ones generated by the first-order $\Sigma\Delta$ quantizer in* (2.4) *with the same input and initial value, say $\{q_i\}_{i=1}^N$.*

Proof. Fix $k \leq N$. We first find a non-recursive formula for q_k. By (2.4) $q_1 = \lfloor x + u_0 \rfloor$ and $u_1 = \langle x + u_0 \rangle$. In turn, $q_2 = \lfloor x + \langle x + u_0 \rangle \rfloor$ and $u_2 = \langle x + \langle x + u_0 \rangle \rangle$. Repeating this argument, one can observe that

$$q_k = \left\lfloor x + \langle \underbrace{x + \langle \dots \langle x}_{(k-1)times} + u_0 \rangle \rangle \dots \rangle \right\rfloor .$$

Now, observing that $\langle \underbrace{x + \langle \dots \langle x}_{(k-1)times} + u_0 \rangle \rangle \dots \rangle = \langle (k-1)x + u_0 \rangle$, we can write $q_k = \lfloor x + \langle (k-1)x + u_0 \rangle \rfloor$. So, in other words:

$$q_k = \begin{cases} 0 & \text{if } \langle (k-1)x + u_0 \rangle \in [0, 1-x), \\ 1 & \text{if } \langle (k-1)x + u_0 \rangle \in [1-x, 1), \end{cases}$$

Since k is arbitrary $\{q_i\}_{i=1}^N$ is identical to $\{b_i\}_{i=1}^N$ as claimed. □

If $x = p/q$ with p, q coprime natural numbers, then it is easy to see that f_{x,u_0} is periodic with minimal period q. If x is irrational then f_{x,u_0} is not ultimately periodic and it is called *Sturmian*.

Definition 3.1.3. [16, p.74] For every word f over an alphabet S and for any natural number $N > 0$, $g_f(N)$ is defined to be $\mathrm{card}(F \cap \{0,1\}^N)$, where F is the set of subwords of f.

In this thesis, the alphabet we will consider is $S = \{0, 1\}$. Then the Sturmian words are exactly the non-ultimately periodic words f such that $g_f(N) = N + 1$. For the following definition, we will stick to the notation used in Mignosi's paper, [16].

Definition 3.1.4. [16, p.73] (*Farey points*) For any natural number $N > 0$, a point $x \in [0, 1]$ is an *N-Farey point* if it is of the form $x = p/q$ with p and q coprime natural numbers, $p \geq 0$, $1 \leq q \leq N$.

We notice that, 0 and 1 are two Farey points for any natural number. Here are some early results derived in [16].

Proposition 3.1.5. *Let $x \in [0,1)$ and f_{x,u_0}, F_{x,u_0} and g_f be defined as above. Then the followings hold.*

(i) [16, Corollary 3, p.73] The set F_{x,u_0} of subwords of f_{x,u_0} depends only on the sequence $\{-ix\},\ i \in \mathbb{N},\ i > 0$, and consequently, depends only on x and does not depend on u_0. So we can assume $u_0 = 0$ and use f_x and F_x from now on.

(ii) [16, Corollary 4, p.73] For any natural number $N > 0$, and for any $x \in (0,1)$, $g_{f_x}(N) \leq N + 1$; if $x = p/q$ with p and q coprime natural numbers, then $g_{f_x}(N) \leq q$.

(iii) [16, Theorem 6, p.74] For each natural number $N > 0$, if $x \in (0,1)$ is not an N-Farey point then $g_{f_x}(N) = N + 1$; if $x \in (0,1)$ is such that $x = p/q$ with p and q coprime natural numbers $g \leq N$, then $g_{f_x}(N) = q$.

Below, f_x denotes $f_{x,0}$, and F_x denotes F_{x,u_0} (because the set F_{x,u_0} does not depend on u_0, we drop the subscript u_0). In [16], Mignosi uses these results to determine the number of distinct length-N subwords of all possible infinite words that could be generated by the scheme given in (3.1). In other words, [16] calculates the cardinality of the set

$$A_N := \bigcup_{x \in [0,1)} F_x \cap \{0,1\}^N.$$

In order to achieve this, Mignosi establishes the following important facts.

Proposition 3.1.6. *Let $N \in \mathbb{N}$ be given. Then the followings hold.*

(i) [16, Corollaries 9-10 and Theorem 11] Let t and u be two consecutive N-Farey points. Then for any $x, y \in (t,u)$ with $x < y$, we have

$$F_x \cap \{0,1\}^N = F_y \cap \{0,1\}^N.$$

Moreover, $F_x \cap \{0,1\}^N \supset F_u \cap \{0,1\}^N$ and $F_x \cap \{0,1\}^N \supset F_t \cap \{0,1\}^N$. (Note that from Proposition 3.1.5, we also know that $g_{f_x} = g_{f_y} = card(F_x \cap \{0,1\}^N) = N + 1$.)

(ii) [16, Corollary 15] If $t < x < u = p/q < y < v$ where t, u, v are three consecutive N-Farey points with p and q coprime natural numbers then $F_x \cap F_y \cap \{0,1\}^N = F_u \cap \{0,1\}^N$ and so card$((F_y \cap \{0,1\}^N) - F_x) = N - q + 1$.

(iii) [16, Theorem 16] If $0 < x < u = p/q < y < v$ where u, v are two consecutive N-Farey points with p and q coprime natural numbers then $((F_y \cap \{0,1\}^N) - F_u) \cap F_x = \emptyset$.

Proposition 3.1.6 is enough to determine the cardinality of A_N. If there are a total of K N-Farey points, including 0 and 1, then there are $K-1$ mutually exclusive subintervals between them. Let us order these N-Farey points and call the subinterval between ith and $(i+1)$st N-Farey points as $T(i)$. Proposition 3.1.6-(i) basically says that all points in set $T(i)$ has the same length-N subword set, which can be denoted as:

$$G_i^N := F_x \cap \{0,1\}^N,\ x \in T(i)$$

whose cardinality is $N+1$. In addition, by Proposition 3.1.6-(ii) the intersection of G_{i-1}^N and G_i^N is the length-N subword set of ith N-Farey point, which can be denoted as:

$$F_i^N := F_y \cap \{0,1\}^N = G_{i-1}^N \cap G_i^N$$

where y is the ith N-Farey point. If the ith N-Farey point can be written as p/q, p, q coprime, then card$(F_i^N) = q$. So we can observe that $A_N = \bigcup_{i<K} G_i^N$.

Finally Proposition 3.1.6-(iii) says that given an interval $T(i)$ (one of the $K-1$ intervals explained above), the length-N subwords introduced by this interval, in addition to the ones produced by previous intervals, have the cardinality of card$(G_i^N \setminus G_{i-1}^N)$. In other words, given i:

$$\text{card}(G_i^N \setminus \bigcup_{j<i} G_j^N) = \text{card}(G_i^N \setminus G_{i-1}^N) = N - q + 1$$

where ith N-Farey point can be written as p/q, p, q coprime.

Cardinality of A_N: Combining all results and arguments above, we can calculate that:

$$\text{card}(A_N) = N + 1 + \sum_{j=2}^{N}(N - j + 1) \times \text{card}(\{p/j \in (0,1) : p, j \text{ coprime}\}) \tag{3.2}$$

It is a fact that $\text{card}(\{p/j \in (0,1) : p, j \text{ coprime}\}) = \varphi(j)$ where φ is the *Euler function*. Using results from [11, p.268], the desired result follows.

Theorem 3.1.7. *card*(A_N) *is asymptotically equivalent to* $\frac{N^3}{\pi^2}$.

3.2 Possible codewords of ΣΔ quantization

Now we want to count all possible codewords of length N that can be generated by a single bit ideal first-order ΣΔ quantizer where x and u_0 are in $[0, 1)$. To this end, we will use the main results from the previous section. Recall that given $x \in [0, 1)$, the set of all length-N subwords of the infinite word generated by x via (3.1) does not depend on the initial value u_0. In the case of ΣΔ quantization, by Proposition 3.1.2, our codewords are the first N bits of these infinite sequences. So given x, different choices of u_0 may give different codewords. Next we will show that the two slightly different formulations, i.e., "fixing u_0 and considering all possible length-N subwords" and "varying u_0 and considering only the first N bits", are in fact equivalent. First we introduce some notation.

Definition 3.2.1. Consider f_{x,u_0} and corresponding $b_n(x, u_0), n \geq 0$ defined in (3.1). Now given $u_0 \in [0, 1)$, define

$$A_{u_0}(N) := \{\{b_i(x, u_0)\}_{i=1}^{N} : x \in [0, 1)\}$$

to be the all possible outcomes of a single bit ideal ΣΔ quantizer with fixed initial value, u_0, and for all $x \in [0, 1)$.

This should not be confused with A_N which was defined in the previous section. A_N was the set of all possible length-N subwords of infinite words

generated by the first-order ΣΔ quantizer (equivalently by (3.1)) for all $x, u_0 \in [0,1)$, whereas $A_{u_0}(N)$ is the set of length-N words that consist of the first N bits generated by the ΣΔ quantizer with fixed u_0 but for all $x \in [0,1)$. We are interested in determining $\text{card}(\bigcup_{u_0\in[0,1)} A_{u_0}(N))$.

Proposition 3.2.2. *Given* $N \in \mathbb{N}$, $\displaystyle\bigcup_{u_0\in[0,1)} A_{u_0}(N) = A_N$

Proof. For simplicity, call $Y := \bigcup_{u_0\in[0,1)} A_{u_0}(N)$. It is obvious that each element of Y also belongs to A_N. Now fix $x \in [0,1)$. Remember that initial value is not important for A_N. We need to prove that each subword of the infinite word $f_{x,0}$, indeed belongs to Y. Say, $f_{x,0} = b_1(x,0)b_2(x,0)\ldots$ as defined in (3.1). Fix an arbitrary natural number $m > 0$ and let $\{b_m, \ldots, b_{m+N-1}\}$ be our arbitrary subword. By (3.1), $b_m = \lfloor x+\langle(m-1)x\rangle\rfloor$. Now choose $u_0 = \langle(m-1)x\rangle$ and consider the word $f_{x,u_0} = a_1(x,u_0)a_2(x,u_0)\ldots$. Since $u_0 \in [0,1)$, $\{a_i\}_{i=1}^N$ clearly belongs to Y. So it is enough to show that

$$\{a_i(x,u_0)\}_{i=1}^N = \{b_i\}_{i=m}^{i=m+N-1}$$

By (3.1) $a_1(x,u_0) = \lfloor x + \langle(m-1)x\rangle\rfloor$ which is equal to b_m. Fix $1 < k \leq N$:

$$\begin{aligned} a_k &= \lfloor x + \langle(k-1)x + \langle(m-1)x\rangle\rangle\rfloor \\ &= \lfloor x + \langle(k+m-2)x\rangle\rfloor \end{aligned}$$

on the other hand:

$$\begin{aligned} b_{m+k-1} &= \lfloor x + \langle(m+k-1-1)x\rangle\rfloor \\ &= \lfloor x + \langle(m+k-2)x\rangle\rfloor \end{aligned}$$

$$\Rightarrow a_k(x,u_0) = b_{m+k-1}(x,0)$$

Since k is arbitrary, our claim becomes true which completes the proof. □

So, we proved the following important fact about first-order ΣΔ schemes

with constant input.

Theorem 3.2.3. *The total number of length-N codewords that can be produced using the first-order $\Sigma\Delta$ scheme in* (2.4)*, regardless of the initial condition $u_0 \in [0,1)$ and the input $x \in [0,1)$ is asymptotically equivalent to N^3/π^2.*

Remark: A weaker version of this result was known in the $\Sigma\Delta$ literature. In particular, in [10] and [12] it is shown that if the initial value of the quantizer is fixed and input $x \in [0,1)$, the number of different codewords is $O(N^2)$. This result will be derived in Section 4.2 of this thesis.

3.3 An alternative comparison between PCM and $\Sigma\Delta$

In this chapter we have shown one of the main observations of this thesis: Number of different codewords that can be generated by an ideal $\Sigma\Delta$ quantizer is only order of $O(N^3)$, more precisely, asymptotically equivalent to $\frac{N^3}{\pi^2}$ while there are 2^N binary words of length N.

Accuracy of first-order $\Sigma\Delta$ schemes: The observation above means that $\{q_i\}_{i=1}^N$, the "raw" codewords obtained by the first-order $\Sigma\Delta$ scheme via (2.4) can be compressed substantially. In particular, since there are a total of N^3/π^2 codewords, we only need N^3 distinct labels (we omit $1/\pi^2$ in the rest of the discussion), thus $R = 3\log_2 N$ bits to encode all these codewords in a lossless fashion. In other words, there is a bijection $\psi_{N,R} : A_N \mapsto \{1, 2, \ldots, 2^R\}$ with $R = 3\log_2 N$. Consequently, we can calculate the "actual accuracy" of the first-order $\Sigma\Delta$ schemes in terms of this new bit rate. In particular, set

$$\widetilde{E}_R^{\Sigma\Delta} := \psi_{N,R} \circ E_N^{\Sigma\Delta}, \quad \widetilde{D}_R^{\Sigma\Delta} := D_N^{\Sigma\Delta} \circ \psi_{N,R}^{-1}.$$

Then we have the following.

Theorem 3.3.1. *In the setting described above, we have*

$$\sup_{x\in[0,1)} |x - \widetilde{D}_R^{\Sigma\Delta}(\widetilde{E}_R^{\Sigma\Delta}(x))| \leq 2^{-R/3},$$

regardless of the initial condition u_0.

Proof. Because $\psi_{N,R}$ is a bijection, we clearly have

$$\widetilde{D}_R^{\Sigma\Delta}(\widetilde{E}_R^{\Sigma\Delta}(x)) = D_N^{\Sigma\Delta}(E_N^{\Sigma\Delta}(x))$$

for all $x \in [0,1)$. We also know that

$$\sup_{x\in[0,1)} |x - D_N^{\Sigma\Delta}(E_N^{\Sigma\Delta}(x))| < 1/N.$$

Substituting $R = 3\log_2 N$ above yields the desired result. □

Since $\psi_{N,R}(A_N)$ can be encoded using R bits, Theorem 3.3.1 shows that, after a post-compression stage (which is lossless), we get the approximation error corresponding to the first-order ΣΔ scheme to be $O(2^{-\frac{R}{3}})$, comparable to $O(2^{-N})$, which is the optimal rate and is achieved by the PCM quantizer (see (2.2)).

Note that for the ΣΔ case, after quantization step we will need to compress the output codewords. This can be done adaptively by using techniques from adaptive coding and data compression.

One of the main reasons ΣΔ schemes are preferred in practice over PCM is their superior robustness properties. It is important to note that a post-compression stage will be performed once the analog input is converted to a digital bitstream. Since the operations in the digital domain can be typically done with perfect accuracy, this compression stage does not influence the robustness properties of the underlying ΣΔ scheme. On the other hand, the total number of possible codewords depend (sensitively) on the actual parameters that are used when implementing the scheme, and may increase if the ideal scheme given in (2.4) is replaced with some non-ideal version. This issue will be discussed in Chapter 4 in detail.

Chapter 4

Error Analysis of $\Sigma\Delta$ Quantization

In this chapter, we first formalize certain types of imperfections that may be considered while designing an encoder. For such imperfections, it is well known that $\Sigma\Delta$ schemes are robust in the sense of Definition 2.1, e.g., [3]. However, in this thesis, our main emphasis is the total number of length-N codewords produced by a $\Sigma\Delta$ scheme, and as alluded to above, this number may depend on the actual parameters that are used in the implementation. In this chapter, we investigate this dependence.

In the previous chapter, we counted all possible codewords that can be generated by an ideal $\Sigma\Delta$ quantizer. Our counting technique took the advantage of considering all possible inputs, x, and initial values, u_0, from the interval $[0, 1)$. But it has to be noted that given a constant input x, the outcome of the quantizer contains information on the initial value u_0. This is because, unlike the case when considering all subwords of the infinite word generated by x, in $\Sigma\Delta$ quantization we deal with the first, say, N bits of the sequence. So in further analysis, we will talk about the effect of u_0 often.

4.1 Constant offset error

First, we consider the "imperfect" first-order $\Sigma\Delta$ quantizer with a constant *offset error* $\delta \in (-\delta_{max}, \delta_{max})$, where $\delta_{max} > 0$ is some fixed margin. A non-ideal quantizer with offset error will have a quantizer threshold of $1 - \delta$.

More precisely, this new quantizer is described by:

$$\begin{aligned} q_n^\delta &= Q(\tilde{u}_{n-1} + x + \delta) \\ \tilde{u}_n &= \tilde{u}_{n-1} + x - q_n^\delta \\ n &= 1, 2, \dots \end{aligned} \tag{4.1}$$

Here the quantizing map Q is as in (2.5), and $\{q_i^\delta\}_{i=1}^N$ is the output vector of the system. It is well known that such an imperfect quantizer still generates approximations to x with approximation error of the order $O(N^{-1})$, i.e., the first-order $\Sigma\Delta$ scheme is robust with respect to constant offset errors.

Next, we investigate the set of possible codewords that can be obtained using the scheme described in (4.1). The following lemma is essentially identical to [10, Lemma 2.1]. Since the underlying $\Sigma\Delta$ scheme in [10] is slightly different than the ones given in this thesis, we will give a somewhat different proof here.

Lemma 4.1.1. *Let $x \in [0, 1)$ and $\delta \in (-\delta_{max}, \delta_{max})$ be fixed. Suppose $\{q_n^\delta\}_{n=1}^N$ is obtained by running (4.1) with some initial condition $\tilde{u}_0$. Then $\{q_n^\delta\}_{n=1}^N$ is identical to the output sequence $\{q_n\}_{n=1}^N$ that is obtained using the ideal $\Sigma\Delta$ quantizer, given in (2.4) quantizer with the initial condition $u_0 := \tilde{u}_0 + \delta$.*

Proof. We will prove this by induction. Our claim is that for all $n \in \mathbb{N}$, $q_n = q_n^\delta$ and $u_n = \tilde{u}_n + \delta$. First set $n = 1$ and note that

$$\begin{aligned} \tilde{u}_1 &= u_0 + x - \underbrace{Q(u_0 + x + \delta)}_{q_1^\delta} + (\delta - \delta) \\ u_1 &= (u_0 + \delta) + x - \underbrace{Q(u_0 + x + \delta)}_{q_1}. \end{aligned}$$

Consequently, our claim is true for $n = 1$. Now assume that the claim is

true for n. Then

$$\tilde{u}_{n+1} = \tilde{u}_n + x + - \underbrace{Q(\tilde{u}_n + x + \delta)}_{q^{\delta}_{n+1}} + (\delta - \delta) \tag{4.2}$$

$$u_{n+1} = u_n + x - \underbrace{Q(u_n + x)}_{q_{n+1}}$$

By our hypothesis, $\tilde{u}_n + \delta = u_n$. It follows that $Q(\tilde{u}_n + x + \delta) = Q(u_n + x)$, and thus $q^{\delta}_n = q_n$. In addition, we can write (4.2) again as:

$$\tilde{u}_{n+1} = u_n + x - Q(u_n + x) - \delta$$

which proves $\tilde{u}_{n+1} = u_{n+1} - \delta$. □

This shows that first-order $\Sigma\Delta$ schemes are insensitive to fixed (but unknown to the both the encoder and the decoder) offset error. To ensure that the number of distinct codewords remain $O(N^3)$, the initial condition u_0 of the ideal scheme should remain in $[0, 1)$. This can be guaranteed by choosing $\tilde{u}_0$, the initial condition of the non-ideal scheme, in $[\delta_{max}, 1-\delta_{max})$. Another issue here is *stability* of this new scheme: we need to ensure that $|\tilde{u}_n|$ remain bounded. We can show that if $\delta_{max} < 1/2$ and $\tilde{u}_0 \in [\delta_{max}, 1-\delta_{max})$, we have

$$-\delta_{max} \leq \tilde{u}_n < 1 + \delta_{max}, \quad \forall n > 0.$$

4.2 Number of codewords with fixed initial value and offset error

In Chapter 3, we showed that the number of all codewords that can be generated by a first-order $\Sigma\Delta$ quantizer is of order $O(N^3)$. In this section, we consider the case of arbitrary but fixed initial value and fixed offset error, and review some results in the literature that provide upper bounds on the number of distinct codewords in this setting. As discussed in the previous section, studying a $\Sigma\Delta$ quantizer with a fixed initial value u_0 and fixed offset error δ reduces to the case of studying the ideal quantizer with initial

value $u_0 + \delta$, i.e., $\Sigma\Delta$ quantizer is insensitive against fixed and bounded offset errors. So we will only consider nonideal quantizers. Let us assume that an arbitrary initial value $u_0 \in [0, 1)$ is given. It is obvious that the number of all possible codewords that can be generated in this setup with a constant input from $[0, 1)$ is order of $O(N^3)$, since this is the total number of codewords over all possible initial values. In fact, it has been well known that the number of length-N codewords for a first-order $\Sigma\Delta$ quantizer with a fixed arbitrary initial value is $O(N^2)$, e.g. [12]. For the special case of $u_0 = 0$, number of distinct codewords of length-N becomes closely related to number of N-Farey points which is asymptotically equivalent to $\frac{3N^2}{\pi^2}$. For this result we stick to the notation in [10] and give a proof for the sake of completenes. Below, $[N]$ denotes the set $\{1, 2, \ldots, N\}$.

Proposition 4.2.1. *[10, Lemma 2.2] Fix $u_0 \in [0, 1)$ and define the sequence $\{s_j\}_{j=1}^{J}$ by ordering every distinct element of the set*

$$S_N(u_0) := \{(z - u_0)/n : z \in [n], n \in [N]\} \ \cup \ \{0, 1\}$$

so that $0 \leq s_1 < s_2 < \ldots s_J \leq 1$. Then all $x \in [s_j, s_{j+1})$, when used as the input of the first-order $\Sigma\Delta$ scheme in (2.4) with the initial condition u_0, result in the same codeword $(q_1, q_2, \ldots . q_n)$. The points s_j are called the associated quantization threshold crossings*.*

Proof. Let $x, u_0 \in [0, 1)$, suppose we run the scheme in (2.4) to obtain u_n and q_n. Then since $x + u_n$ remains in $[0, 2)$ for all n (this follows from the stability of the first-order $\Sigma\Delta$ scheme) and since for $u \in [0, 2)$, $Q(u) = \lfloor u \rfloor$, we can rewrite (2.4) (with the given initial condition u_0) as

$$\begin{aligned} q_n &= \lfloor u_{n-1} + x \rfloor \\ u_n &= \langle u_{n-1} + x \rangle \end{aligned}$$

Then by a series of recursive substitution as in (2.10), we have

$$u_n = u_0 + nx - \sum_{i=1}^{n} q_i \tag{4.3}$$

On the other hand, using $u_n = \langle u_{n-1} + x \rangle$, we can check that

$$\begin{aligned} u_1 &= \langle u_0 + x \rangle \\ u_2 &= \langle u_1 + x \rangle = \langle \langle u_0 + x \rangle + x \rangle = \langle u_0 + 2x \rangle \\ &\vdots \\ u_n &= \langle u_0 + nx \rangle \end{aligned} \tag{4.4}$$

Combining the two formulas (4.3) and (4.4), we get:

$$Z_n := \sum_{i=1}^{n} q_i = \lfloor u_0 + nx \rfloor \tag{4.5}$$

It can be shown that the sequence $\{q_i\}_{i=1}^n$ is uniquely determined by the sequence $\{Z_i\}_{i=1}^n$, and *vice versa* (see [10, Appendix A]). Since Z_n changes value at the points $\{x = (j - u_0)/n : j \in [n]\}$ (at $x = 0$, $Z_n = 0$ for all n and at $x = 1$, $Z_n = n$ for all n). So distinct intervals $[s_j, s_{j+1})$ given by the (ordered) threshold points in S_N correspond to distinct N-tuples $\{Z_i\}_{i=1}^N$ and thus to distinct length-N codeword $\{q_i\}_{i=1}^N$. So, the cardinality of distinct codewords generated by this quantizer will be equal to the cardinality of the set S_N. □

It can be observed that cardinality of $S_N(u_0)$ can be at most $\frac{N(N+1)}{2} + 1$ which is of order $O(N^2)$. For the special case of $u_0 = 0$, some of the values are repeated, so $S_N(0)$ becomes the set of N-Farey points, i.e., $\{j/n \in [0, 1] : \gcd(j, n) = 1\}$ and its cardinality is asymptotically equivalent to $\frac{3N^2}{\pi^2} \approx 0.304N^2$ as $N \to \infty$. It can be also proven that if u_0 is irrational, every distinct (j, n) pair produces a distinct point in S_N, and consequently the number of distinct codewords attains its maximum. It is important to note that for each distinct value of u_0, the corresponding set of $O(N^2)$ output codewords may be different (depending on u_0). Besides, we know that total number of all possible codewords with $u_0 \in [0, 1)$ is order of $O(N^3)$ (in fact asymptotically equivalent to N^3/π^2). It is interesting to observe that an uncountable union of sets, each with $O(N^2)$ elements has a cardinality that

is only of order $O(N^3)$.

4.3 Varying offset error

Another imprecision type we will consider is a nonideal quantizer with offset error changing with n independently, i.e., put δ_n in place of δ in (4.1) where the only information we have on δ_n is that $|\delta_n| < \delta_{max}$ for all n where δ_{max} is an appropriate margin. For now, we will assume that $\delta_{max} < 1/4$. Clearly, this is a harder case to analyze since we can no longer compensate the effect of offset error by changing the initial value. Instead, we need to do a finer analysis. We first give the recursion that describes this new "imprecise" first-order $\Sigma\Delta$ scheme:

$$\begin{aligned} \hat{q}_n &= Q(\hat{u}_{n-1} + x + \delta_n) \\ \hat{u}_n &= \hat{u}_{n-1} + x - \hat{q}_n, \\ n &= 1, 2, \dots \end{aligned} \tag{4.6}$$

where the quantizing map Q is as in (2.5) and $\delta_n \in (-\delta_{max}, \delta_{max})$ are arbitrary. Our goal again is to count the number of codewords that can be generated by this quantizer. It is non-trivial to relate this question to the results given in Chapter 3 where we considered all possible initial values and constant inputs from $[0, 1)$ with the ideal scheme. Our results in Section 4.2, where we counted distinct codewords when u_0 is fixed and when we had a constant offset error, do not generalize to the varying offset error case. Here, we will again fix $\hat{u}_0$, arbitrarily chosen from some interval, and we seek the total number of possible output codewords over all admissible input values. In this case (i.e., when $\hat{u}_0$ is fixed and the offset error is varying), the value of $\hat{u}_0$ and δ_{max} have a significant effect on the upper bounds for the number of codewords.

4.3.1 Non-recursive formulation

The scheme given in (4.6) is a recursive formula, and therefore it is difficult to analyze. In order to determine the quantization thresholds, we first obtain a non-recursive formulation as we have in the proof of Proposition 4.2.1. It is easy to see that

$$\hat{u}_n = \hat{u}_0 + nx - \sum_{i=1}^{n} \hat{q}_i \tag{4.7}$$

Note that this is essentially identical to (4.3). In order to get a relation similar to (4.4), we will assume that *quantizing map* Q behaves as a *floor function*. For that assumption to be true, $\hat{u}_{n-1} + x + \delta_n$ should remain in the interval $[0, 2)$. Then we can do the following induction:

$$\begin{aligned}
\hat{u}_1 &= \hat{u}_0 + x - \lfloor \hat{u}_0 + x + \delta_1 \rfloor + (\delta_1 - \delta_1) \\
&= \langle \hat{u}_0 + x + \delta_1 \rangle - \delta_1 \\
\hat{u}_2 &= \hat{u}_1 + x - \lfloor \hat{u}_1 + x + \delta_2 \rfloor + (\delta_2 - \delta_2) \\
&= \langle \hat{u}_1 + x + \delta_2 \rangle - \delta_2 \\
&= \langle \langle \hat{u}_0 + x + \delta_1 \rangle - \delta_1 + x + \delta_2 \rangle - \delta_2 \\
&= \langle \hat{u}_0 + 2x + \delta_2 \rangle - \delta_2 \\
&\vdots \\
\Rightarrow \quad \hat{u}_n &= \langle \hat{u}_0 + nx + \delta_n \rangle - \delta_n
\end{aligned} \tag{4.8}$$

Combining (4.7) and (4.8), we get:

$$\begin{aligned}
\langle \hat{u}_0 + nx + \delta_n \rangle - \delta_n &= \hat{u}_0 + nx - \sum_{i=1}^{n} \hat{q}_i + (\delta_n - \delta_n) \\
\Rightarrow \quad \hat{Z}_n := \sum_{i=1}^{n} \hat{q}_i &= \lfloor \hat{u}_0 + nx + \delta_n \rfloor
\end{aligned} \tag{4.9}$$

As it can be observed in (4.9), $\hat{Z}_i$ is only affected by the offset error that occurs at the ith step, i.e., δ_i. Also, an argument as in the proof of Proposition 4.2.1 shows that the sequence $\{\hat{Z}_i\}_{i=1}^{n}$ uniquely determines the output

codeword $\{\hat{q}_i\}_{i=1}^n$. For further analysis, we will take advantage of (4.9), but first we state the conditions on the ranges of $\hat{u}_0$ and x in order to have a *stable* system and thus to be able to use the argument above.

Lemma 4.3.1. *Consider the* $\Sigma\Delta$ *scheme in* (4.6). *Let* $|\delta_n| < \delta_{max}$ *for all* $n \geq 1$ *and* $\delta_{max} < 1/4$. *If* $x \in (2\delta_{max}, 1 - 2\delta_{max})$ *and* $\hat{u}_0 \in [0, 1)$, *then* $\hat{u}_n + x + \delta_{n+1} \in [0, 2)$ *for all* $n \geq 1$ *and* $\hat{u}_n$ *remains bounded with* $\hat{u}_n \in [-\delta_{max}, 1 + \delta_{max})$ *for all* $n > 0$.

Proof. It is trivial that $\hat{u}_n + x + \delta_{n+1} \in [0, 2)$ holds for $n = 0$. By using the relation $\hat{u}_{n+1} = \hat{u}_n + x - Q(\hat{u}_n + x + \delta_{n+1})$ for $n = 0$, we can see that $\hat{u}_1 \in [-\delta_{max}, 1 + \delta_{max})$ and $\hat{u}_1 + x + \delta_2 \in [0, 2)$. Since the claim holds for $n = 0$ and $n = 1$, by the nature of recursive algorithm of the scheme, it will hold for all $n > 0$. □

4.3.2 Threshold regions

To find quantization thresholds of the scheme given in (4.6), we use the equation (4.9) as we used (4.5) in the ideal quantizer case. It can be observed that when $\hat{u}_0 + nx + \delta_n$ is a natural number, $\hat{Z}_n$ changes value, which uniquely corresponds to a change in $\hat{q}_n$. Since we have an independent offset errors δ_n, in this setting instead exact threshold points, we have threshold regions.

Definition 4.3.2. Consider the nonideal N-bit $\Sigma\Delta$ quantizer with varying offset error $\delta := \{\delta_1, \ldots, \delta_N\}$ where $\delta_n \in (-\delta_{max}, \delta_{max})$ for $n \geq 1$ and constant input $x \in (2\delta_{max}, 1 - 2\delta_{max})$. Let $\hat{u}_0 \in [0, 1)$ be fixed. Define the map $t_{\hat{u}_0}(z, n, \delta)$ and interval $I_{\hat{u}_0}(z, n)$ for $z \in [n], n \in [N]$, as follows:

$$t_{\hat{u}_0} : (z, n, \delta) \to \frac{z - \hat{u}_0 - \delta_n}{n} \tag{4.10}$$

$$I_{\hat{u}_0}(z, n) := \left(\frac{z - \hat{u}_0 - \delta_{max}}{n}, \frac{z - \hat{u}_0 + \delta_{max}}{n} \right) \tag{4.11}$$

Then we can define the set of *threshold regions* and *points* as follows:

$$R_N(\hat{u}_0) := \{I_{\hat{u}_0}(z, n) : I_{\hat{u}_0}(z, n) \cap (2\delta_{max}, 1 - 2\delta_{max}) \neq \emptyset, z \in [n], n \in [N]\}$$
$$T_N(\hat{u}_0, \delta) := \{t_{\hat{u}_0}(z, n, \delta) : I_{\hat{u}_0}(z, n) \in R_N(\hat{u}_0), z \in [n], n \in [N]\}$$

$R_N(\hat{u}_0)$ is a set of intervals $I_{\hat{u}_0}(z,n)$, each of which has at least one quantization threshold point, $t_{\hat{u}_0}(z,n,\delta)$. Observe that $R_N(\hat{u}_0)$ does not depend on the sequence $\{\delta_n\}_{n=1}^N$ whereas $T_N(\hat{u}_0,\delta)$, i.e., the exact locations of the threshold points, depends on $\{\delta_n\}_{n=1}^N$. All points belonging to the interval between two consecutive quantization threshold points, generate one and the same codeword. Since the domain of input is not $[0,1)$ but $[2\delta_{max}, 1-2\delta_{max})$ anymore, before considering the affect of varying input, in the following lemma, we will derive the number of distinct codewords if the threshold point $t_{\hat{u}_0}(z,n,\delta)$ was in the middle of the interval $I_{\hat{u}_0}(z,n)$, i.e., if $\delta_n = 0$ for all n.

Definition 4.3.3. Consider the nonideal N-bit $\Sigma\Delta$ quantizer with offset threshold δ_{max} and initial value $\hat{u}_0$.

(i) The interval $(2\delta_{max}, 1-2\delta_{max})$ is called the *admissable region* and a pair (z,n) is called as *admissable pair* if $I_{\hat{u}_0}(z,n) \in R_N(\hat{u}_0)$.

(ii) We define the set of *original quantization thresholds* as follows:

$$\hat{S}_N(\hat{u}_0) := \{\frac{(z-\hat{u}_0)}{n} : (z,n) \text{ is admissable pair}\}$$

Notation: Observe that $\hat{S}_N(\hat{u}_0) = T_N(\hat{u}_0, 0)$, where $\delta = 0$ means $\delta_n = 0$ for all $n > 0$. Similarly $|\delta| < \delta_{max}$ means $|\delta_n| < \delta_{max}$ for all n.
In the rest of discussion, we omit $\hat{u}_0$ in the notation when there is no chance of confusion.

Lemma 4.3.4. *Consider the ideal N-bit $\Sigma\Delta$ quantizer with constant input $x \in (2\delta_{max}, 1-2\delta_{max})$. Let $\hat{u}_0 \in [0,1)$ and $\delta_{max} < 1/4$. Then card$(\hat{S}_N(\hat{u}_0)) \asymp N^2$.*

Proof. First observe that $\text{card}(\hat{S}_N(\hat{u}_0)) < \text{card}(S_N(\hat{u}_0)) \leq \frac{N(N+1)}{2} + 1$ since *admissable region* is smaller than $[0,1)$. It is enough to count the integer values $\lfloor \hat{u}_0 + nx \rfloor$ can take. Fix $1 \leq n_0 \leq N$. The range is: $2n_0\delta_{max} < \hat{u}_0 + n_0 x < n_0 + 1 - 2n_0\delta_{max}$. If $\delta_{max} < 1/4$, then $n_0 + 1 - 2n_0\delta_{max} > 2n_0\delta_{max}$. Number of integer values in this range depends on δ_{max} and can be calculated

to be $n_0 - 2\lfloor 2n_0\delta_{max}\rfloor$. Now consider the sequence $\left\{\lfloor\frac{k}{2\delta_{max}}\rfloor\right\}_{k=0}^{2N\delta_{max}}$. We can observe that if $\lfloor\frac{k}{2\delta_{max}}\rfloor < n_0 < \lfloor\frac{k+1}{2\delta_{max}}\rfloor$, then:

$$n_0 - 2\lfloor 2n_0\delta_{max}\rfloor = n_0 - 2k$$

So we can write the following equalities:

$$\begin{aligned}
\mathrm{card}(\hat{S}_N(\hat{u}_0)) &= 1 + \sum_{k=0}^{2N\delta_{max}-1} \sum_{i=\lfloor\frac{k}{2\delta_{max}}\rfloor+1}^{\lfloor\frac{k+1}{2\delta_{max}}\rfloor} (i-2k) \\
&= 1 + \sum_{k=0}^{2N\delta_{max}-1} \sum_{i=\lfloor\frac{k}{2\delta_{max}}\rfloor+1}^{\lfloor\frac{k+1}{2\delta_{max}}\rfloor} i - 2\sum_{k=0}^{2N\delta_{max}-1} \sum_{i=\lfloor\frac{k}{2\delta_{max}}\rfloor+1}^{\lfloor\frac{k+1}{2\delta_{max}}\rfloor} k \\
&= 1 + \sum_{i=1}^{N} i - 2\sum_{k=0}^{2N\delta_{max}-1} \left(k \sum_{i=\lfloor\frac{k}{2\delta_{max}}\rfloor+1}^{\lfloor\frac{k+1}{2\delta_{max}}\rfloor} 1 \right)
\end{aligned}$$

Observe that:

$$\frac{1}{2\delta_{max}} - 2 < \sum_{i=\lfloor\frac{k}{2\delta_{max}}\rfloor+1}^{\lfloor\frac{k+1}{2\delta_{max}}\rfloor} 1 < \frac{1}{2\delta_{max}}$$

Using the right inequality, it follows:

$$\mathrm{card}(\hat{S}_N(\hat{u}_0)) > 1 + \sum_{i=1}^{N} i - 2\frac{1}{2\delta_{max}} \left(\sum_{k=0}^{2N\delta_{max}-1} k \right)$$

From here, it can be calculated that $\mathrm{card}(\hat{S}_N(\hat{u}_0)) \gtrsim N^2(0.5 - 2\delta_{max})$ as $N \to \infty$. This shows that $O(N^2)$ is also a lower bound for $\mathrm{card}(\hat{S}_N(\hat{u}_0))$. □

But if the orientation of threshold points with respect to themselves change due to a different error sequence, intervals between them will produce different codewords. We can formulate the set that we want to count with a similar notation to Definition 3.2.1, as follows:

Definition 4.3.5. Consider the nonideal N-bit $\Sigma\Delta$ quantizer with initial

value $\hat{u}_0 \in [0,1)$ be fixed. Let $\hat{A}_{\hat{u}_0}^{\delta_{max}}(N)$ denote the set of all possible N-bit output codewords, $\{\hat{q}_i\}_{i=1}^N$, that can be generated by the scheme (4.6) for all possible inputs $x \in (2\delta_{max}, 1-2\delta_{max})$ and $\{\delta_n\}_{n=1}^N$ sequences within the bounds $|\delta_n| < \delta_{max}$.

The cardinality of $\hat{A}_{\hat{u}_0}^{\delta_{max}}(N)$ depends on δ_{max} and $\hat{u}_0$. If every threshold point was in the middle of corresponding threshold region, i.e., if $\delta_n = 0$ for all n, then we would have the number of codewords to be order of $O(N^2)$. Next, we will investigate the affect of varying offset error on the number of codewords.

Definition 4.3.6. Consider the quantization thresholds and regions given in Definition 4.3.2 for a fixed $\hat{u}_0 \in [0,1)$. Assume that there is a subset $\mathcal{D}$ of $R_N(\hat{u}_0)$ with k elements, say $\mathcal{D} = \{I_{\hat{u}_0}^j\}_{j=1}^k$, which has the following properties:

$$\bigcap_{1\le j\le k} I_{\hat{u}_0}^j \neq \emptyset$$

$$\left(\bigcap_{1\le j\le k} I_{\hat{u}_0}^j\right) \cap I_{\hat{u}_0} = \emptyset \quad \text{for all } I_{\hat{u}_0} \in (R_N(\hat{u}_0) \setminus \mathcal{D})$$

Such a set $\mathcal{D}$ will be referred to as a *k-degree intersection*.

Definition 4.3.7. Assume that $\hat{u}_0 \in [0,1)$, N is fixed, and we have a sequence of errors $\delta = \{\delta_n\}_{n=1}^N$. Let us label the original quantization thresholds in the increasing order as:

$$t_1 \le t_2 \le \ldots t_{l-1} \le t_l$$

such that $T_N(\hat{u}_0, 0) = \{t_1, \ldots, t_l\}$ where $l = \text{card}(T_N(\hat{u}_0, 0))$. Note that $t_i = t(z, n, 0)$ for some (z, n) pair and $i \in [l]$. For a given input $x \in (2\delta_{max}, 1 - 2\delta_{max})$ we define the binary word $b_{x,\delta}(i)$ by:

$$b_{x,\delta}(i) := \begin{cases} 0 & \text{if } x < t_{\hat{u}_0}(z,n,\delta) \text{ and } t_i = t_{\hat{u}_0}(z,n,0) \\ 1 & \text{if } x \ge t_{\hat{u}_0}(z,n,\delta) \text{ and } t_i = t_{\hat{u}_0}(z,n,0) \end{cases} \quad i = 1, 2, \ldots, l$$

Remark: The original threshold points t_i may change their orientation with respect to each other when an offset error is introduced. But we keep labelling them with respect to their initial order in the original case. When x and δ are fixed, the word $b_{x,\delta}$ actually tells the location of x with respect to perturbed threshold points. Observe that the word $b_{x,\delta}$ remains fixed if x varies in the interval between two consecutive threshold points.

Proposition 4.3.8. *Assume that N and $\hat{u}_0 \in [0,1)$ are fixed. Consider the set of all possible binary words $b_{x,\delta}$ for all $x \in (2\delta_{max}, 1-2\delta_{max})$ and $|\delta| < \delta_{max}$:*

$$B_{\hat{u}_0}^{\delta_{max}} := \bigcup_{x,\delta} b_{x,\delta}$$

Then there is a bijection between $B_{\hat{u}_0}^{\delta_{max}}$ and $\hat{A}_{\hat{u}_0}^{\delta_{max}}(N)$.

Proof. Suppose for some x, δ, we have $b_{x,\delta} = b$. Then define $\mathcal{L}(x) := \{t(z,n,\delta) \leq x : (z,n) \text{ is admissable pair}\}$ which consists of all quantization threshold points that remain to the left of x (when δ is given). Note that $\mathcal{L}(x)$ is completely and uniquely determined by b (i.e., different b gives different $\mathcal{L}(x)$). On the other hand, the set $\mathcal{L}(x)$ uniquely determines the sequence $\{\hat{Z}_n\}_{n=1}^N$ which was defined to be the partial sum of a codeword, i.e., $\hat{Z}_n = \sum_{i=1}^n \hat{q}_i$. Indeed, fix $n \in [N]$, and define $z_0 := \max\{z : t(z,n,\delta) \in \mathcal{L}(x)\}$. If $\{z : t(z,n,\delta) \in \mathcal{L}(x)\} = \emptyset$, set $z_0 = 0$. Then, $t(z_0,n,\delta) \leq x < t(z_0+1,n,\delta)$, and this implies $\hat{Z}_n = z_0$. To see this, recall that $\hat{Z}_n = \lfloor \hat{u}_0 + nx + \delta_n \rfloor$ and the points where $\hat{Z}_n$ changes value are actually the quantization threshold points given in (4.10). Since n is arbitrary, given the set $\mathcal{L}(x)$ the sequence $\{\hat{Z}_n\}_{n=1}^N$ can be determined. In addition, a different $\mathcal{L}(x)$ would produce a different $\{\hat{Z}_n\}_{n=1}^N$ sequence since any change in the orientation of a given point x with respect to quantization points (being on the left or right of them) affects the digits of $\hat{Z}_n$. Finally, as there is a one-to-one correspondence between $\{\hat{Z}_n\}_{n=1}^N$ and $\{\hat{q}_n\}_{n=1}^N$ sequences, we conclude that whenever word b is given, a codeword $\{\hat{q}_n\}_{n=1}^N$ is uniquely determined, i.e., different b corresponds to different $\{\hat{q}_n\}_{n=1}^N$. This implies that $\text{card}(B_{\hat{u}_0}^{\delta_{max}}) \leq \text{card}(\hat{A}_{\hat{u}_0}^{\delta_{max}}(N))$. A similar argument can be used in order to show the opposite of this result. In other words, given a codeword $\{\hat{q}_n\}_{n=1}^N$ for some x, δ, there is a

unique binary word b that corresponds to this codeword. Changing the codeword would result in getting a different binary word. This implies that $\text{card}(B_{\hat{u}_0}^{\delta_{max}}) \geq \text{card}(\hat{A}_{\hat{u}_0}^{\delta_{max}}(N))$. So we conclude that cardinalities of two finite sets $B_{\hat{u}_0}^{\delta_{max}}$ and $\hat{A}_{\hat{u}_0}^{\delta_{max}}(N)$ are equal which completes the proof. □

Theorem 4.3.9. *Let $\hat{u}_0 \in [0, 1)$, N and δ_{max} be fixed in the scheme* (4.6)*. If there exists a k-degree intersection set $\mathcal{D}$, then card$(\hat{A}_{\hat{u}_0}^{\delta_{max}}(N))$ is at least 2^k.*

Proof. Let us say that original threshold points are $\{t_j\}_{j=1}^{l}$ in the increasing order and we know that l is order of $O(N^2)$. Now assume that for some index set Λ with cardinality k, $\mathcal{D} = \{I^j\}_{j\in\Lambda}$ where I^j is the threshold region that has the center at t_j. By definition of *k-degree intersection* we know that, points with indices in Λ can change order with respect to themselves in every way possible. This implies that the restriction $b_{x,\delta} \mid_\Lambda$ can have every possible combination of binary sequences with length k, for some x value and δ sequence. Since there are 2^k such binary sequences, and at least 2^k different $b_{x,\delta}$ sequences, we conclude that there are as many different codewords, $\{q_n\}_{n=1}^{N}$ that belong to $\hat{A}_{\hat{u}_0}^{\delta_{max}}(N)$. This completes the proof. □

Theorem 4.3.9 gives an exponential order for the number of codewords that can be generated by a single *k-degree intersection*. For a given $k \leq N$, existence and the number of *k-degree intersections* depend on the initial value $\hat{u}_0$ and δ_{max}. Because the former determines the location of the quantization threshold regions and the latter determines the size of them. For instance if $\hat{u}_0 = 0$ or $\hat{u}_0 = 0.5$, then $N/2$ quantization threshold regions are centered at 0.5 due to points $\{z/n : n = 2z, n \leq N\}$. Then no matter δ_{max} is, we have at least one *$\frac{N}{2}$-degree intersection*. This is not desired since we would need too many bits to represent output codewords of order $O(2^{N/2})$. In the ideal case, we took advantage of relatively small number of distinct words that could be generated, so we could make a reasonable comparison between approximation rates of $\Sigma\Delta$ quantization and PCM method. But in the case of varying offset error, this comparison can easily fail for certain u_0 and δ_{max} values. So we need to derive some conditions on these values in

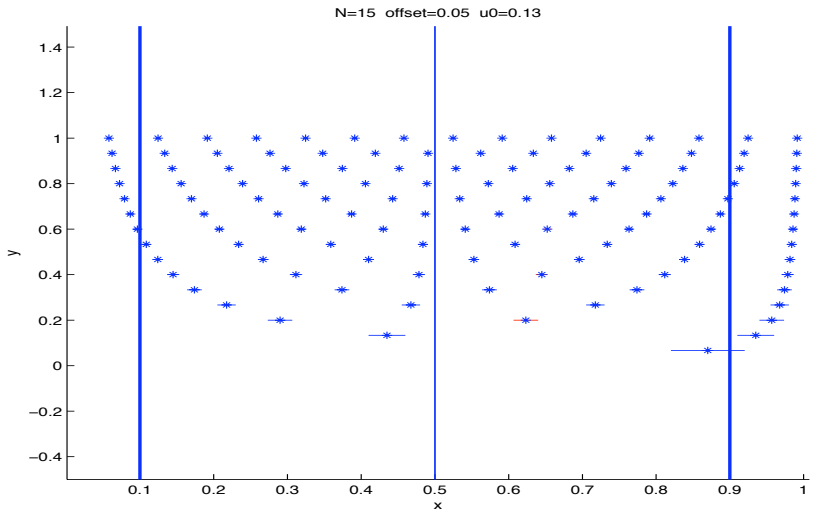


Figure 4.1: Quantization threshold regions, with levels depending on n

order not to have any *(k-degree) intersections.*

Definition 4.3.10. Let $\hat{u}_0 \in [0,1)$, $N \in \mathbb{N}$ and $\delta_{max} < 1/4$ be given. Assume that n_0 and n_1 are given with $n_0 + m = n_1 \leq N$, $m > 0$. If there exist $z_0 \leq n_0$ and $z_1 \leq n_1$ such that $I_{\hat{u}_0}(z_0, n_0) \cap I_{\hat{u}_0}(z_1, n_1) \neq \emptyset$, then we say that there is an *m-level intersection.* If $I_{\hat{u}_0}(z_0, n_0)$ and $I_{\hat{u}_0}(z_1, n_1)$ belong to $R_N(\hat{u}_0)$ then we say that *intersection is in the admissable region.*

For an easier visualization of intersections defined above, see Figure 4.1. Each quantization threshold region $I_{\hat{u}_0}(z, n_0)$ is drawn at the level $y = n_0/N$ and at each level there are n_0 intervals. Center of the interval $I_{\hat{u}_0}(z, n)$ is located at $(\frac{z-\hat{u}_0}{n})$ and its size is $\frac{2\delta_{max}}{n}$. Vertical lines on the left and right are the boundaries for *admissable region*, with equations $x = 2\delta_{max}$ and $x = 1 - 2\delta_{max}$.

Definition 4.3.11. Let us be given two intervals I and J. We say that I is on the left of J if for all $x \in I, y \in J$, we have $x < y$. Denote it as $I < J$. If I is on the left of J, then J is on the right of I, i.e., $J > I$.

Remark: If $n = n_0$ is fixed:

$$\frac{z - \hat{u}_0 + \delta_{max}}{n_0} < \frac{z + 1 - \hat{u}_0 - \delta_{max}}{n_0} \Leftrightarrow \delta_{max} < 1/2$$

that is, there is no intersection of threshold regions on the level n_0, i.e., $I_{\hat{u}_0}(n_0, z) < I_{\hat{u}_0}(n_0, z+1)$ for all $z \in [n_0 - 1]$ and $\delta_{max} < 1/4$. Even though we assume that exact locations of threshold points are independent from each other, actually for a fixed $n = n_0$, threshold points $t_{\hat{u}_0}(z, n_0, \delta)$ all depend on the same δ_{n_0} value, see (4.10). Since we just showed that threshold regions on the same level n_0 do not intersect, for the purposes of counting all possible intersections of threshold regions, we can assume that they are independent.

Lemma 4.3.12. *Consider the N-bit nonideal $\Sigma\Delta$ scheme with $\hat{u}_0 \in [0, 1)$ and $\delta_{max} < 1/4$. Let m, n be given such that $1 \leq m < n \leq N$. Then:*

$$\begin{aligned} &\forall z \in [n] : \\ &\quad I_{\hat{u}_0}(z, n) \in R_N(\hat{u}_0) \Rightarrow I_{\hat{u}_0}(z, n) < I_{\hat{u}_0}(z, n - m) \end{aligned}$$

Similarly:

$$\begin{aligned} &\forall z \in [n - m] : \\ &\quad I_{\hat{u}_0}(z + m, n) \in R_N(\hat{u}_0) \Rightarrow I_{\hat{u}_0}(z, n - m) < I_{\hat{u}_0}(z + m, n) \end{aligned}$$

Proof. For the first claim, assume that $I_{\hat{u}_0}(z, n) \in R_N(\hat{u}_0)$ for a given $z \in [n]$. This implies that the right end of the interval $I_{\hat{u}_0}(z, n)$ should be greater than $2\delta_{max}$, i.e.,

$$\begin{aligned} &\frac{z - \hat{u}_0 + \delta_{max}}{n} > 2\delta_{max} \Leftrightarrow \\ &z > (2n - 1)\delta_{max} + \hat{u}_0 \end{aligned} \tag{4.12}$$

Simple algebra shows that (4.12) implies:

$$\begin{aligned} & z > (2n-1)\delta_{max} + \hat{u}_0 \Rightarrow \\ & z > \delta_{max}\left(\frac{2n}{m} - 1\right) + \hat{u}_0 \Leftrightarrow \\ & \frac{z - \hat{u}_0 - \delta_{max}}{n-m} > \frac{z - \hat{u}_0 + \delta_{max}}{n} \end{aligned} \tag{4.13}$$

(4.13) means that the right end of $I_{\hat{u}_0}(z,n)$ is smaller than the left end of $I_{\hat{u}_0}(z,n-m)$, i.e., $I_{\hat{u}_0}(z,n) < I_{\hat{u}_0}(z,n-m)$. Then first claim becomes true for all $z \in [n]$.
Similarly for the second claim, assume that $I_{\hat{u}_0}(z+m,n) \in R_N(\hat{u}_0)$ for a given $z \in [n-m]$. Then the left end of the interval $I_{\hat{u}_0}(z+m,n)$ should be smaller than $1-2\delta_{max}$, i.e.,

$$\begin{aligned} & \frac{z + m - \hat{u}_0 - \delta_{max}}{n} < 1 - 2\delta_{max} \Leftrightarrow \\ & z < n + \hat{u}_0 - m - (2n-1)\delta_{max} \end{aligned} \tag{4.14}$$

By simple algebra, (4.14) implies:

$$\begin{aligned} & z < n + \hat{u}_0 - m - (2n-1)\delta_{max} \Rightarrow \\ & z < n + \hat{u}_0 - m - \delta_{max}\left(\frac{2n}{m} - 1\right) \Leftrightarrow \\ & \frac{z - \hat{u}_0 + \delta_{max}}{n-m} < \frac{z + m - \hat{u}_0 - \delta_{max}}{n} \end{aligned} \tag{4.15}$$

(4.15) means that the right end of $I_{\hat{u}_0}(z,n-m)$ is smaller than the left end of $I_{\hat{u}_0}(z+m,n)$, i.e., $I_{\hat{u}_0}(z,n-m) < I_{\hat{u}_0}(z+m,n)$. Then second claim is also true for all $z \in [n-m]$. This completes the proof. □

In fact, this lemma says that, in order to have an *(m-level) intersection* in the *admissable region*, two threshold intervals have to have z values, say z_0, z_1, with property $1 \leq |z_0 - z_1| \leq m-1$.

Corollary 4.3.13. *Consider the N-bit nonideal ΣΔ scheme with $\hat{u}_0$ and δ_{max}. Then there is no 1-level intersection in the admissable region.*

Proof. If we choose $m = 1$ in the Lemma 4.3.12, then it is straightforward. □

In the following lemma, we will give a condition in order not to have any *(m-level) intersection* in the *admissable region.*

Lemma 4.3.14. *Consider the N-bit nonideal $\Sigma\Delta$ scheme with $0 \leq \hat{u}_0 < 1$ and δ_{max}. Let $1 < m < N$ be given. If there is no integer value in the interval*

$$\left[\frac{j}{m} + \hat{u}_0 - \delta_{max}\left(\frac{2N}{m} - 1\right), \frac{j}{m} + \hat{u}_0 + \delta_{max}\left(\frac{2N}{m} - 1\right)\right] \tag{4.16}$$

for all $j \in \mathbb{N}$ such that $0 \leq j \leq m$, then there is no m-level intersection in the admissable region.

Proof. Fix $n > m$. Then most generally consider two threshold intervals $I_{\hat{u}_0}(z+j, n)$ and $I_{\hat{u}_0}(z, n-m)$ that may possibly have an *m-level intersection*, where $1 \leq j \leq m-1$ and $1 \leq z \leq n-m$. These bounds for j and z are implied by Lemma 4.3.12. The condition for these two intervals not to have an *m-level intersection* is either $I_{\hat{u}_0}(z+j, n) < I_{\hat{u}_0}(z, n-m)$ or $I_{\hat{u}_0}(z+j, n) > I_{\hat{u}_0}(z, n-m)$. If we write these conditions explicitly:

$$\frac{z + j - \hat{u}_0 + \delta_{max}}{n} < \frac{z - \hat{u}_0 - \delta_{max}}{n - m}$$

or

$$\frac{z + j - \hat{u}_0 - \delta_{max}}{n} > \frac{z - \hat{u}_0 + \delta_{max}}{n - m}$$

If we simplify these inequalities, we have the condition:

$$z \notin \left[\frac{jn}{m} - j + \hat{u}_0 - \delta_{max}\left(\frac{2n}{m} - 1\right), \frac{jn}{m} - j + \hat{u}_0 + \delta_{max}\left(\frac{2n}{m} - 1\right)\right] \tag{4.17}$$

Note that it is sufficient to have:

$$z \notin \left[\frac{jn}{m} - j + \hat{u}_0 - \delta_{max}\left(\frac{2N}{m} - 1\right), \frac{jn}{m} - j + \hat{u}_0 + \delta_{max}\left(\frac{2N}{m} - 1\right)\right] \tag{4.18}$$

as these new sets include the sets given in (4.17) for every n and j. Recall that z is a positive integer. So, if there is no integer value in the interval in (4.18) for all $m < n \leq N$ and $1 \leq j \leq m-1$, then there is no *m-level intersection* for the whole set $R_N(\hat{u}_0)$. Since we can write the equivalent intervals in "mod 1", it is equal to say that there is no integer value in the interval:

$$\left[\left\langle \frac{jn}{m} \right\rangle - j + \hat{u}_0 - \delta_{max}\left(\frac{2N}{m} - 1\right), \left\langle \frac{jn}{m} \right\rangle - j + \hat{u}_0 + \delta_{max}\left(\frac{2N}{m} - 1\right)\right]$$

Since j is an integer, we can omit it. So we have the interval:

$$\left[\left\langle \frac{jn}{m} \right\rangle + \hat{u}_0 - \delta_{max}\left(\frac{2N}{m} - 1\right), \left\langle \frac{jn}{m} \right\rangle + \hat{u}_0 + \delta_{max}\left(\frac{2N}{m} - 1\right)\right]$$

Lastly, we claim that for a fixed m,

$$\left\{\left\langle \frac{jn}{m} \right\rangle : 1 \leq j \leq m-1, m < n \leq N\right\} \subseteq \left\{\frac{j}{m} : 0 \leq j \leq m-1\right\}$$

Say $jn = am + b$ where $b < m$ and $a, b \in \mathbb{N}$. It follows that:

$$\frac{jn}{m} = a + \frac{b}{m} \Rightarrow \left\langle \frac{jn}{m} \right\rangle = \frac{b}{m}$$

So for every $1 \leq j < m$, there is $0 \leq b < m$ that satisfies $\left\langle \frac{jn}{m} \right\rangle = \frac{b}{m}$. Note that $b = 0$ is also possible for the remainder if n is a multiple of m. If we choose $n = m + 1$, then $\frac{jn}{m} = \frac{j}{m}$, which proves our claim. It is sufficient to consider:

$$\left[\frac{j}{m} + \hat{u}_0 - \delta_{max}\left(\frac{2N}{m} - 1\right), \frac{j}{m} + \hat{u}_0 + \delta_{max}\left(\frac{2N}{m} - 1\right)\right]$$

where $0 \leq j \leq m-1$ and $1 < m < n \leq N$. We can include the case $j = m$ since it is identical to the case $j = 0$ as long as we are interested in the integer values in the interval. This completes the proof. □

4.4 Some conditions on initial value and δ_{max}

In order to do post-processing compression for a $\Sigma\Delta$ quantizer, number of possible codewords should be bounded reasonably. For instance having a bound with exponential order will not help much for our practical purposes in comparison to other quantization schemes. As we saw earlier, an *k-degree intersection* is enough to make number of codewords at least order of $O(2^k)$. Deriving conditions on δ_{max} and $\hat{u}_0$ in order not to have any *k-degree intersection* does not seem easy. Instead, we will use the Lemma 4.3.14 to derive conditions for δ_{max} in order not to have any *m-level intersection* for $1 < m < N$.

Theorem 4.4.1. *Consider the N-bit nonideal $\Sigma\Delta$ quantizer scheme given in* (4.6) *with initial value $\hat{u}_0 \in [0, 1)$. If $\delta_{max} < \frac{1}{2N^2-2}$, then there exists some values for $\hat{u}_0$ that guarantee there is no m-level intersection for all m such that $1 < m < N$, so $\hat{A}^{\delta_{max}}_{\hat{u}_0}(N) = \hat{A}^{0}_{\hat{u}_0}(N)$ and has cardinality order of $O(N^2)$ as the ideal case.*

Proof. We can generalize the result of Lemma 4.3.14 in order to have a sufficient condition for not having any *m-level intersection* for $1 < m < N$.

$$z \notin \left[\frac{j}{m} + \hat{u}_0 - \delta_{max}\left(\frac{2N}{m} - 1\right), \frac{j}{m} + \hat{u}_0 + \delta_{max}\left(\frac{2N}{m} - 1\right)\right] \tag{4.19}$$

$\forall z \in \mathbb{Z},\ 1 < m < N,\ 0 \leq j \leq m$. So we again came across Farey series with this condition. We have some intervals around $\hat{u}_0$-shifted $(N-1)$-Farey points. We can write these shifted points as:

$$\mathcal{F}_{N-1}(\hat{u}_0) := \{\frac{j}{m} + \hat{u}_0 : \gcd(j, m) = 1, 1 < m < N, 0 \leq j \leq m\}$$

Now, it can be observed that only three natural numbers that can be contained by intervals around $\mathcal{F}_{N-1}(\hat{u}_0)$, are 0, 1 and 2 for small δ_{max}. Furthermore, if we choose δ_{max} small enough so that the intervals around any two consecutive points in $\mathcal{F}_m(\hat{u}_0)$ do not intersect, then for some $\hat{u}_0$, (4.19) should be satisfied. Note that, if $\frac{c}{a}$ and $\frac{d}{b}$ are two consecutive $(N-1)$-Farey

points, then their difference is $\frac{1}{ab}$. We can write the condition on δ_{max} as:

$$\delta_{max}\left(\frac{2N}{a}-1\right)+\delta_{max}\left(\frac{2N}{b}-1\right)<\frac{1}{ab}\Leftrightarrow$$
$$\delta_{max}[2N(a+b)-2ab]<1 \tag{4.20}$$

for all $a, b \leq m < N$ and $n \leq N$. It can be showed that $\max(2N(a+b)-2ab)$ is attained at $a = b = N-1$. Then we have the following upper bound for δ_{max} which guarantees no *m-level intersection*:

$$\delta_{max} < \frac{1}{2N^2-2}$$

which is order of $O(N^{-2})$ bound. We can find the appropriate $\hat{u}_0$ by the following formulas:

$$\left.\begin{aligned}\hat{u}_0+\frac{c}{a}+\delta_{max}\left(\frac{2N}{a}-1\right)<1\\ \hat{u}_0+\frac{d}{b}-\delta_{max}\left(\frac{2N}{b}-1\right)>1\end{aligned}\right\}\Rightarrow$$
$$1-\frac{d}{b}+\delta_{max}\left(\frac{2N}{b}-1\right)<\ \hat{u}_0<1-\frac{c}{a}-\delta_{max}\left(\frac{2N}{a}-1\right) \tag{4.21}$$

for any two consecutive $(N-1)$-Farey points, $\frac{c}{a}$ and $\frac{d}{b}$. If we substitute δ_{max} by the worst case bound of $\frac{1}{2N^2-2}$ in (4.21), we will have a nonempty interval to choose for $\hat{u}_0$. □

Lemma 4.4.1 says if δ_{max} is small enough, for some discrete values of $\hat{u}_0$, possible codewords of a nonideal $\Sigma\Delta$ quantizer remain same as the ideal case. Yet, one may seek for a continuous relation between δ_{max} and $\hat{u}_0$ in order to have a sufficient condition for keeping the same output codewords set, i.e., $\hat{A}^{\delta_{max}}_{\hat{u}_0}(N) = \hat{A}^{0}_{\hat{u}_0}(N)$. Let us assume that for a particular value of $\hat{u}_0$, 1 falls between two consecutive points of set $\mathcal{F}_{N-1}(\hat{u}_0)$, say $\frac{c}{a}+\hat{u}_0$ and $\frac{d}{b}+\hat{u}_0$. Here 0 and 1 belong to $(N-1)$-Farey points and it will be enough to consider only 1 as the possible natural number z that can be included by some interval. This is because Farey points and intervals around them are

symmetrical with respect 0.5 and shifting by $\hat{u}_0$ is always to the right. So there is a circular relation when we consider the possible natural numbers 0,1 and 2 for z. Then, let us have the particular constraint:

$$1 - \frac{d}{b} \leq \hat{u}_0 < 1 - \frac{c}{a}$$

If $\hat{u}_0$ is under the above constraint, then in order not to have intersection of two intervals around the chosen points $\frac{c}{a} + \hat{u}_0$ and $\frac{d}{b} + \hat{u}_0$, we need following conditions:

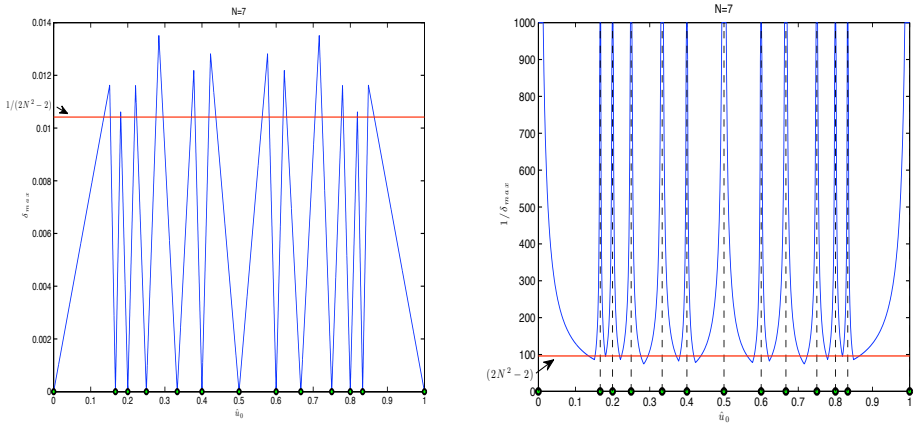
$$\left.\begin{aligned} 1 - \hat{u}_0 - \frac{c}{a} > \delta_{max}\left(\frac{2N}{a} - 1\right) \;&\Rightarrow\; \frac{a(1-\hat{u}_0) - c}{2N - a} > \delta_{max} \\ \hat{u}_0 + \frac{d}{b} - 1 > \delta_{max}\left(\frac{2N}{b} - 1\right) \;&\Rightarrow\; \frac{b(\hat{u}_0 - 1) + d}{2N - b} > \delta_{max} \end{aligned}\right\} \tag{4.22}$$

We also need to guarantee that any consecutive two interval will not intersect by setting $\delta_{max} < \frac{1}{2N^2-2}$. Then we have the final relation:

$$\delta_{max} < \min\left\{\frac{a(1-\hat{u}_0) - c}{2N - a}, \frac{b(\hat{u}_0 - 1) + d}{2N - b}, \frac{1}{2N^2 - 2}\right\}$$

whenever $1 - \frac{d}{b} \leq \hat{u}_0 < 1 - \frac{c}{a}$. Observe that the relation between $\hat{u}_0$ and δ_{max} is linear between two consecutive $(N-1)$-Farey points and δ_{max} will have an optimum value in that region. As we saw earlier, we expect this optimum value to be order of $O(N^{-2})$. (See Figure 4.2)

Remark: In Figure 4.2-(a) we have triangles with bases located between two consecutive $(N-1)$-Farey points. The linearity of the relation between $\hat{u}_0$ and corresponding δ_{max} comes from the formulas given in (4.22). Peak points of the triangles are where these two formulas become equal. We can observe that δ_{max} values that correspond to these peak points cannot be smaller than $\frac{1}{2N^2-2}$ since this value guarantees that there is no mutual intersection of all intervals given in (4.19), whereas the bounds for δ_{max} in (4.22) guarantees only that there is no intersection of intervals that are around two consecutive $\hat{u}_0$-shifted-Farey points. This observation will be



(a) (b)

Figure 4.2: (a) Linear relation between initial value $\hat{u}_0$ and δ_{max} and (b) order of δ_{max} $(1/\delta_{max})$ with respect to $\hat{u}_0$. N is fixed to 7 for this example. Marked points on the x axis are 6-Farey points.

important for a further discussion in Section 5.2.

Finally, given N, let us write the function Δ_N that gives the formula between $\hat{u}_0$ and δ_{max} in order not to have any *m-level intersections* for $m \leq N$:

$$\Delta_N(\hat{u}_0) = \left\{ \min \left\{ \frac{a(1-\hat{u}_0)-c}{2N-a}, \frac{b(\hat{u}_0-1)+d}{2N-b}, \frac{1}{2N^2-2} \right\} \right.$$
$$\left. : \frac{c}{a}, \frac{d}{b} \text{ are two consecutive } (N-1)\text{-Farey points and } 1-\frac{d}{b} \leq \hat{u}_0 < 1-\frac{c}{a} \right\}$$

Now for a given δ_{max} value, define the set $V_N(\delta_{max}) := \{\hat{u}_0 : \Delta_N(\hat{u}_0) < \delta_{max}\}$. It is important to observe that for all $\hat{u}_0 \in V_N(\delta_{max})$, the non-ideal $\Sigma\Delta$ quantizer (4.1) gives the same codeword of length N as the ideal quantizer in (2.4) would give. We can formulate this as follows: Given δ_{max}

$$\bigcup_{\hat{u}_0 \in V_N(\delta_{max})} A_{\hat{u}_0}^{\delta_{max}}(N) \subset A_N$$

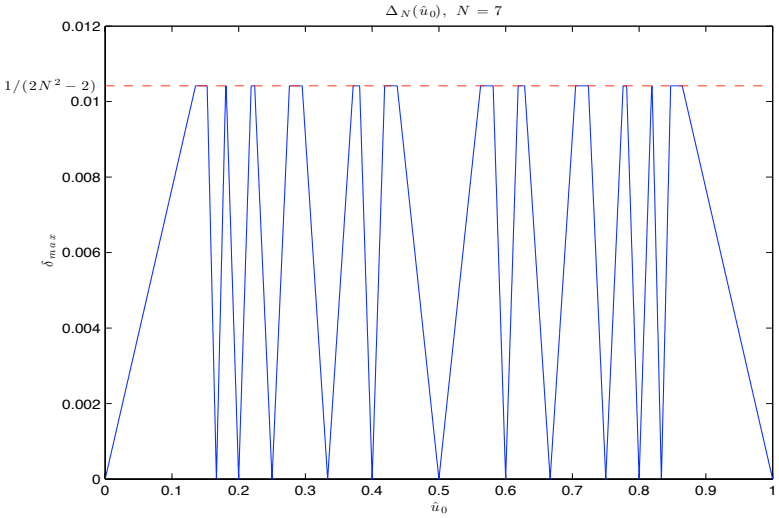


Figure 4.3: Graph of function $\Delta_N(\hat{u}_0)$ for $N = 7$. Dashed line shows the level of $\frac{1}{2N^2-2}$. Marked points on the x axis are 6-Farey points.

and A_N has the cardinality of order $O(N^3)$. In Figure 4.3, it can be seen that $\Delta_N(\hat{u}_0)$ consists of truncated triangles at level $\frac{1}{2N^2-2}$.

Chapter 5

Implications for Bandlimited Functions

In Chapter 1, we mentioned about two basic steps of A/D conversion process, one of which was sampling of functions. Sampling is the reduction of a continuous function to a discrete set of values. The question of how to sample a function in order to make it possible to reconstruct the original function completely was answered for *bandlimited functions* by *Classical (Nyquist-Shannon) Sampling Theorem.* This theorem states that an Ω-bandlimited function f, i.e., $\text{supp}\hat{f} \subseteq [-\Omega, \Omega]$, where $\hat{f}$ is the Fourier transform of f, is completely characterized by sampling it at the Nyquist frequency $\frac{\Omega}{\pi}$, [3].

$$f(t) = \sum_{n\in\mathbb{Z}} f\left(\frac{n\pi}{\Omega}\right) \text{sinc}\left(t - \frac{n\pi}{\Omega}\right) \tag{5.1}$$

where $\text{sinc}(x) = x^{-1}\sin x$. In particular, at any fixed t_0, $f(t)$ is an infinite weighted average of the sample values $f\left(\frac{n\pi}{\Omega}\right)$. For computational purposes, we would want that only a few of these "weights", i.e., $\text{sinc}\left(t - \frac{n\pi}{\Omega}\right)$, to be "large". However except at integer cases (i.e., when $\frac{n\pi}{\Omega} - t_0$ is integer for all n), these weights decay only like $\frac{1}{n}$. In addition, if the samples are not know exactly and we have $f\left(\frac{n\pi}{\Omega}\right) + \varepsilon_n$ with all $|\varepsilon_n| < \varepsilon$, then the corresponding approximation $\tilde{f}_n$ may differ appreciably from $f(t)$. Thus the reconstruction is not local. For a more detailed argument of this we refer to [3], [6] and [4].

However, we can solve this problem by *oversampling* the function at rate higher than Nyquist, say at rate $\frac{\lambda\Omega}{\pi}$, $\lambda > 1$. Our more closely spaced samples are $\{f\left(\frac{n}{\lambda}\right)\}_{n\in\mathbb{Z}}$. Then *sinc kernel* in (5.1) can be replaced by another kernel φ where $\hat{\varphi}(\xi) = 1$ for $|\xi| \leq \Omega$ and $\hat{\varphi}(\xi) = 0$ for $|\xi| \geq \lambda\Omega$. If we choose φ

such that $\hat{\varphi}$ is smooth, we will have the desired fast decay properties. Then we have the formula:

$$f(t) = \sum_{n\in\mathbb{Z}} \frac{1}{\lambda} f\left(\frac{n\pi}{\lambda\Omega}\right) \varphi\left(t - \frac{n\pi}{\lambda\Omega}\right) \tag{5.2}$$

For simplicity we can choose $\Omega = \pi$ in the following arguments which makes sampling rate equal to λ.

5.1 Basic error estimates for PCM and ΣΔ schemes

By oversampling, we bought the freedom of choosing our reconstruction kernel and having a little number of "significantly contributing" samples. While this process turns a continuous time signal to a discrete time signal, samples are still real numbers. We need to quantize this numbers with a given bit budget. In general, if we replace $f(\frac{n}{\lambda})$ by $\tilde{f}(\frac{n}{\lambda}) + \varepsilon_n$ in (5.2) with $|\varepsilon_n| < \varepsilon$, then [3] gives the approximation error as:

$$|f(t) - \tilde{f}(t)| \leq \varepsilon \frac{1}{\lambda} \sum_{n\in\mathbb{Z}} \left|\varphi\left(t - \frac{n}{\lambda}\right)\right| \leq \varepsilon C_\varphi \tag{5.3}$$

where $C_\varphi := \lambda^{-1}\|\varphi'\|_{L^1} + \|\varphi\|_{L^1}$.

PCM schemes: If we have a bit budget of N for each sample, the simplest way to do this is finding the truncated binary expansion of the sample. We can assume that $|f(t)| \leq 1$ for all t. Then each sample $f(t)$ is reconstructed with error 2^{-N}, i.e., $\varepsilon \lesssim 2^{-N}$. This leads to $|f(t) - \tilde{f}(t)| \leq C2^{-N}$ where C is independent of N and f. This type of quantization is used in representing audio signals, but they have implementation drawbacks which we discussed in Chapter 2 in details. Mainly, designing a circuit which implements binary expansion of a real valued sample is not very robust with respect to imperfections. In practice, ΣΔ schemes has been preferred widely with their robustness properties compared to PCM schemes.

ΣΔ schemes: The oversampling ratio λ indicates the number of samples we have in a Nyquist interval. In ΣΔ schemes, instead of using N-bit per sample, we use them to quantize all λ samples in a Nyquist interval. Here we actually set $N = \lambda$ and choose $\lambda \gg 1$. Then we have the corresponding formula for ΣΔ scheme as:

$$\begin{aligned} u_n &= u_{n-1} + f\left(\frac{n}{\lambda}\right) - q_n^\lambda \\ q_n^\lambda &= Q\left(u_{n-1} + f\left(\frac{n}{\lambda}\right)\right) \end{aligned} \tag{5.4}$$

with an initial condition $u_0 \in [0, 1)$ and Q as defined in (2.5). In circuit implementation, the range of n in (5.4) is $n \geq 1$. [3] shows that u_n and q_n^λ can be written directly in terms of u_{n+1} and f_{n+1} when $n < 0$. But in the rest of the discussion about ΣΔ schemes, we will assume that $n \geq 1$.
Next, if we use the following reconstruction formula:

$$\tilde{f}(t) = \frac{1}{\lambda} \sum_n q_n^\lambda \varphi\left(t - \frac{n}{\lambda}\right)$$

then [3] derives the following approximation error estimate:

$$|f(t) - \tilde{f}(t)| \leq \frac{1}{\lambda} \|\varphi'\|_{L^1} \tag{5.5}$$

We discussed before that this polynomial decay $\frac{1}{\lambda}$ is rather poor with respect to PCM schemes but main advantage of ΣΔ schemes was their robustness, i.e., keeping this approximation rate same even when implemented with imprecise (nonideal) quantizers. If we replace the quantization formula in (5.4) with imprecise version:

$$q_n^\lambda = Q\left(u_{n-1} + f\left(\frac{n}{\lambda}\right) + \delta_n\right)$$

where $|\delta_n| < \delta$, then [4] derives an error estimate for imprecise ΣΔ scheme as follows:

$$|f(t) - \tilde{f}(t)| \leq \frac{C_{\delta,\varphi}(1+\delta)}{\lambda}$$

This shows that ΣΔ scheme is robust since it keeps the same approximation rate, $\frac{1}{\lambda}$ in the imperfect case. If the first-order ΣΔ scheme is generalized to higher-order schemes, one can get better bounds for approximation error. [3] derives the approximation rate to be order of $\frac{1}{\lambda^k}$ for a general kth order ΣΔ scheme. But this is still poor in comparison to PCM scheme.

5.2 A new ΣΔ quantization setup for bandlimited functions

We saw that the approximation rate of a ΣΔ quantizer is not affected by possible imperfections. While we have this robustness, we want to be able to do post-processing compression owing to the fact that number of distinct codewords of length N produced by a first-order ΣΔ quantizer is much lesser than 2^N. In Chapter 3, we showed that number of distinct codewords for an ideal quantizer is order of $O(N^3)$ which leads to an approximation rate order of $O(2^{-\frac{N}{3}})$ (see Section 3.3). This result is valid for a constant input quantizer and this fact gives a motivation to us to come up with following simple but somewhat interesting sampling technique for a first-order ΣΔ quantizer.

A new setup: Assume that we oversample our continuous function f with an oversampling rate λ. But instead of feeding our ΣΔ quantizer with values $f(\frac{n}{\lambda})$, we repeat each value N times before skipping to the next one. So the input sequence becomes:

$$\{\ldots, \underbrace{f\left(\frac{n}{\lambda}\right), \ldots, f\left(\frac{n}{\lambda}\right)}_{N-times}, \underbrace{f\left(\frac{n+1}{\lambda}\right), \ldots, f\left(\frac{n+1}{\lambda}\right)}_{N-times}, \ldots\}$$

Then we can write the following recovering formula:

$$\begin{aligned} f(t) &= \frac{1}{\lambda}\sum_n f\left(\frac{n}{\lambda}\right)\varphi_\lambda\left(t-\frac{n}{\lambda}\right) \\ &= \frac{1}{\lambda}\frac{1}{N}\sum_{j=1}^{N}\sum_n f\left(\frac{n}{\lambda}\right)\varphi_\lambda\left(t-\frac{n}{\lambda}\right) \end{aligned}$$

for $\lambda > 1$. If we define $g_n := f\left(\frac{\lceil n/N \rceil}{\lambda}\right)$ and $\phi_\lambda\left(t-\frac{n}{\lambda}\right) := \varphi_\lambda\left(t-\frac{\lceil n/N \rceil}{\lambda}\right)$, we have the following formula:

$$f(t) = \frac{1}{N\lambda}\sum_n g_n \phi_\lambda\left(t-\frac{n}{\lambda}\right)$$

Observe that ϕ has desired smoothness properties as φ. So in our new setup, we can assume that we use the input sequence $\{g_n\}_{n\geq 1}$ for the ideal quantizer in (5.4) with an initial condition $u_0 \in [0,1)$. Then we can bound the approximation error with $\frac{1}{N\lambda}$ by using a similar argument as in [3].
If the produced bit sequence is $\{q_n^\lambda\}_{n\geq 1}$, set $\tilde{f}(t) = \frac{1}{N\lambda}\sum_n q_n^\lambda \phi_\lambda\left(t-\frac{n}{\lambda}\right)$ and observe that $g_n - q_n^\lambda = u_n - u_{n-1}$. Then it follows as:

$$\begin{aligned} |f(t)-\tilde{f}(t)| &= \frac{1}{N\lambda}\left|\sum_n (u_n-u_{n-1})\phi_\lambda\left(t-\frac{n}{\lambda}\right)\right| \\ &= \frac{1}{N\lambda}\left|\sum_n u_n\left(\phi_\lambda\left(t-\frac{n}{\lambda}\right)-\phi_\lambda\left(t-\frac{n+1}{\lambda}\right)\right)\right| \\ &\leq \frac{1}{N\lambda}\sum_n\left|\phi_\lambda\left(t-\frac{n}{\lambda}\right)-\phi_\lambda\left(t-\frac{n+1}{\lambda}\right)\right| \\ &\leq \frac{1}{N\lambda}\sum_n\int_{t-\frac{n+1}{\lambda}}^{t-\frac{n}{\lambda}}|\phi'(y)|dy = \frac{1}{N\lambda}\|\phi'\|_{L^1} \end{aligned}$$

A similar result can be derived for the nonideal case in order to show robustness.

Remark: One interesting result of this setup is, by only repeating sample values N times, we have a better approximation rate of $\frac{1}{N\lambda}$. The sampling

rate is kept same. Another result is: This setup allows us to further compress the bit stream q_n^λ. Assuming that $n \geq 1$, N-bit sequence $\{q_1^\lambda, q_2^\lambda, \ldots, q_n^\lambda\}$ can be seen as the output of the ΣΔ scheme given in (2.4) with constant input g_1 and initial value u_0, then the N-bit sequence $\{q_{N+1}^\lambda, q_{N+2}^\lambda, \ldots, q_{2N}^\lambda\}$ can be seen as the output of (2.4) with constant input g_2 and initial value u_N, and so on. We know that for an ideal quantizer all these N-bit codewords belong to the set of A_N with cardinality of order $O(N^3)$ regardless of the initial value, so we can follow the argument of further compression made in Section 3.3.

Nonideal quantizer in the new setup: In the case of a nonideal quantizer with a varying offset error $|\delta_n| < \delta_{max}$, the eligible initial values depend on the δ_{max} value with function $\Delta_N(\hat{u}_0)$ introduced in Section 4.4. More specifically assume that δ_{max} is fixed; if the initial value of the nonideal quantizer is in the set $V_N(\delta_{max})$ then the N-bit codeword will be in the set of original codewords generated by the ideal quantizer with cardinality of order $O(N^3)$. In our setup, the input sequence $\{g_n\}_{n\geq 1}$ changes value at $n = kN + 1$, $k \in \mathbb{N}$. So we can think as we have a constant input quantizer that generates codewords length-N with the initial value sequence of $\{u_{kN}\}_{k\in\mathbb{N}}$. We know from [8, Section 3.1] that u_{kN} has uniform distribution in $[0, 1)$ for all k independently from each other. Since the initial value u_{kN} determines whether N-bit codeword generated with input $g_{(kN+1)}$ is in the set of order $O(N^3)$, it is possible to calculate the probability for having reasonably bounded number of codewords after quantization.

Lemma 5.2.1. *Let δ_{max} be given. For a random $u_0 \in [0, 1)$,*

$$\mathbb{P}(u_0 \in V_N(\delta_{max})) > \frac{\tau - \delta_{max}}{\tau}$$

where $\tau = \frac{1}{2N^2-2}$.

Proof. Let μ be the Lebesgue measure. It is easy to see that $\mathbb{P}(u_0 \in V_N(\delta_{max})) = \mu(\{u_0 : \Delta_N(u_0) < \delta_{max}\})$. At the end of the Section 4.4, we remarked that the peak points of the triangles formed by function $\Delta_N(u_0)$

are not smaller than τ. If we assume that we have such triangles with their peaks exactly at the level τ, and if the corresponding function was $\tilde{\Delta}_N(u_0)$, then from triangle similarity, we would have $\mu(\{u_0 : \tilde{\Delta}_N(u_0) < \delta_{max}\}) = \frac{\tau-\delta_{max}}{\tau}$. Since our original triangles are taller than these ones, we have $\mu(\{u_0 : \Delta_N(u_0) < \delta_{max}\}) > \mu(\{u_0 : \tilde{\Delta}_N(u_0) < \delta_{max}\})$. This proves the lemma. □

Theorem 5.2.2. *Let δ_{max} and an arbitrary bandlimited function f with Nyquist rate 1 be given. Assume that we will oversample function f at rate $\lambda \gg 1$ with the ΣΔ quantizer explained above (it produces N-bit codewords per sample). If an L-unit portion of the function is processed via quantizer with $L \gg 1$, then the probability of having at least k codewords in the set A_N is:*

$$\geq \binom{L\lambda}{k} p^k (1-p)^{L\lambda-k}$$

where $p = \frac{\tau-\delta_{max}}{\tau}$ and $\tau = \frac{1}{2N^2-2}$.

Proof. In the L-unit interval of f, there are $L\lambda$ samples. Each sample is fed into quantizer as a constant input and generates a length-N codeword. Lemma 5.2.1 says that probability of each of these codewords to be in the set A_N is at least $\frac{\tau-\delta_{max}}{\tau}$ and we know that these probabilities are independent. By a simple combinatorics argument we can get the desired result. □

Remark: Theorem 5.2.2 gives us freedom of choosing k such that we can bound the total number of distinct codewords with any order. For instance, if we choose $k = L\lambda - N^3$, then it is certain that k many codewords will be in the set A_N which is of order $O(N^3)$. On the other hand, if we assume that all other $L\lambda - k = N^3$ codewords are out of this set, which is the worst case, we will have total number of $O(N^3) + N^3$ distinct codewords. This is still order of $O(N^3)$.

Lastly we can talk about the post-processing compression in this setup. For a bandlimited function, number of bits used to represent all samples in a Nyquist interval is important. If we oversample with rate λ and use N bits per sample, so it makes λN bits per Nyquist interval. Approximation rate

was $\frac{1}{\lambda N}$ for this ΣΔ quantizer. As a further argument, if we know that there are only order of $O(N^3)$ distinct codewords that may be generated, then we can spend only $R = 3\log_2 N$ bits to represents all codewords, and this leads to an approximation rate of $\frac{1}{\lambda N} = \frac{1}{\lambda 2^{\frac{R}{3}}} = \frac{2^{-\frac{R}{3}}}{\lambda}$. This new setup for ΣΔ quantizers allows us to repeat the argument given in Section 3.3 for bandlimited functions and make a much better comparison between PCM and ΣΔ schemes in terms of approximation rates.

Bibliography

[1] Jean Berstel. Recent results on sturmian words. In *Developments in Language Theory*, pages 13–24, 1995.

[2] J.C. Candy. A use of double integration in sigma delta modulation. *IEEE Trans. Commun.*, COM-33:249–258, Mar. 1985.

[3] Ingrid Daubechies and Ronald A. DeVore. Reconstructing a bandlimited function from very coarsely quantized data: A family of stable sigma-delta modulators of arbitrary order. *Annals of Mathematics*, 158(2):679–710, 2003.

[4] Ingrid Daubechies, Ronald A. DeVore, C. Sinan Güntürk, and Vinay A. Vaishampayan. A/d conversion with imperfect quantizers. *IEEE Transactions on Information Theory*, 52(3):874–885, 2006.

[5] Ingrid Daubechies, C. Sinan Güntürk, Y. Wang, and Özgür Yılmaz. The golden ratio encoder. *CoRR*, abs/0809.1257, 2008.

[6] Ingrid Daubechies and Özgür Yılmaz. Robust and practical analog-to-digital conversion with exponential precision. *IEEE Transactions on Information Theory*, 52(8):3533–3545, 2006.

[7] R.M. Gray. Oversampled sigma-delta modulation. *IEEE Trans. Commun.*, COM-35:481–489, May 1987.

[8] C. Sinan Güntürk. *Harmonic analysis of two problems in signal quantization and compression.* PhD thesis, Princeton Univ., Princeton, NJ, 2000.

[9] C. Sinan Güntürk. One-bit sigma-delta quantization with exponential accuracy. *Comm. Pure. Appl. Math.*, 56(511):1608–1630, 2003.

[10] C. Sinan Güntürk, J. C. Lagarias, and Vinay A. Vaishampayan. On the robustness of single-loop sigma-delta modulation. *IEEE Transactions on Information Theory*, 47(5):1735–1744, 2001.

[11] G.H. Hardy and E.M. Wright. *An Introduction to the Theory of Numbers.* Oxford University Press, 5th edition, 1983.

[12] S. Hein, K. Ibraham, and A. Zakhor. New properties of sigma-delta modulators with dc inputs. *Communications, IEEE Transactions on*, 40(8):1375–1387, Aug 1992.

[13] S. Hein, K. Ibraham, and A. Zakhor. *Sigma Delta modulators: nonlinear decoding algorithms and stability analysis.* Dordrecht, The Netherlands:Kluwer, 1993.

[14] H. Inose and Y. Yasuda. A unity bit coding method by negative feedback. *Proc. IEEE*, 51:1524–1535, Nov. 1963.

[15] D.F. Hoschele Jr. *Analog-to-Digital and Digital-to-Analog Conversion Techniques.* New York:Wiley, 1994.

[16] Filippo Mignosi. On the number of factors of sturmian words. *Theor. Comput. Sci.*, 82(1):71–84, 1991.

[17] R. Schrier S.R. Norsworthy and Eds G.C. Temes. *Delta-Sigma data converters: theory, design, and simulation.* Piscataway, NJ: IEEE Press, 1996.

[18] V.M. Tikhomirov and A.N. Kolmogorov. ε-entropy and ε-capacity of sets in function spaces. *Amer. Math. Soc. Transl.*, 17(2):277–364, 1961.

[19] Özgür Yılmaz. Stability analysis for several sigma-delta methods of coarse quantization of bandlimited functions. *Constructive Approximation*, 18(4):599–623, 2002.