

# Signal Processing for Communications

---

## Lab Assignment 1: Analog to Digital Conversion

### 1 Sampling and quantization

In this assignment we explore different aspects of the A/D conversion process. As you know, this process consists of two different operations: sampling (discretization of time) and quantization (discretization of amplitudes). The sampling operation is linear and, provided that no aliasing takes place, does not incur in any loss of information. The quantization operation, on the other hand, is inherently nonlinear and results in an information loss (it is impossible to know what the value of the unquantized sample was if only its quantized version is available).

Throughout the assignment, we will make use of the Matlab function `quanti.m`:

```
xq = quanti( x, FS, Nbits );
```

where `x` is a vector with the input samples, `xq` is the vector of output quantized values, `FS` is the full-scale value, and `Nbits` is the number of bits in the quantizer. Therefore, the number of quantization levels is  $2^{\text{Nbits}}$ , and the quantization step is  $\text{LSB} = \text{FS}/2^{\text{Nbits}-1}$ . The lowest and highest quantization levels are respectively  $-\text{FS}$  and  $\text{FS}-\text{LSB}$ .

#### □ Task 1

To visualize the effect of the quantizer, it is useful to represent its output in terms of the input, that is, its "input/output curve". For example, for a 2-bit quantizer with  $\text{FS} = 5$ , you may use:

```
x = linspace(-7,7,1000); xq = quanti(x,5,2); plot(x,x,'b',x,xq,'r'); grid on
```

- Repeat for a 4-bit quantizer with the same full-scale value.
- Give your interpretation of the resulting graphs. Do the quantization levels correspond with the values you had expected?
- For both cases, represent the quantization error as a function of input amplitude in the range  $[-7, +7]$  and comment on your results. Is this error always within the  $[-\frac{\Delta}{2}, +\frac{\Delta}{2}]$  interval?

■

	$A = 0.5 \cdot \text{FS}$		$A = 0.75 \cdot \text{FS}$		$A = \text{FS}$		$A = 1.03 \cdot \text{FS}$	
	SQNR (dB)		SQNR (dB)		SQNR (dB)		SQNR (dB)	
$N$	theory	measured	theory	measured	theory	measured	theory	measured
12								
10								
8								
6								
4								

Table 1: Pertaining to Task 2.

We'll start by analyzing quantization for sinusoids<sup>1</sup>. Let  $x(t) = A \cos(2\pi f_0 t)$  be the analog signal to be sampled and quantized. Assume a converter with  $\text{FS} = 5$  and sampling frequency  $f_s = 100$  MHz. For the input signal, let  $f_0 = 18.17$  MHz.

## □ Task 2

- Generate  $15 \cdot 2^{10} = 15 \cdot 1024$  samples of  $x(t)$  and quantize them to  $N = 10$  bits. Assume a full-scale sinusoidal input.

Using the command `hist(x-xq, 40)`, plot the histogram of the quantization error. Do you observe what you expected, or not?

- Explain the operation of the Matlab command `var`. Estimate the variance of the quantization error using `var`, and compare it to its theoretical value. Estimate the value (in dB) of the Signal-to-Quantization Noise Ratio (SQNR) and compare it to its theoretical value (1). Comment on your results.
- Repeat the previous steps for sinusoids with different amplitudes, and with decreasing resolutions of 12, 10, 8, 6 and 4 bits, in order to fill Table 1, rounding the SQNR values (in dB) to two decimal places. Comment on your results.

Explain your results.

■

## 2 Converter overload

From the previous task, it should be clear that whenever the input signal exceeds the full-scale range of the ADC, severe distortion due to clipping is likely to take place. The following task explores the SQNR degradation produced by clipping.

<sup>1</sup>The short application note [1] may help you refresh and clarify some important concepts.

### □ Task 3

- Suppose that you have an  $N$ -bit A/D converter with tunable FS, and you know that your input samples follow a symmetric triangular pdf in some interval  $[-x_0, x_0]$ . Intuitively, how would you set the FS value of your converter? What would the resulting rms value  $\sigma_x$  in dBFS be<sup>2</sup>?
- Explain how to generate in Matlab samples of a random variable following a symmetric triangular pdf with zero mean and rms value  $\sigma_x$ . Check the histogram and use the commands `mean` and `var` to validate your approach.
- Take  $10 \cdot 2^{10}$  of these triangularly distributed samples, quantize them, and estimate the SQNR empirically<sup>3</sup> for  $N = 3, 4, 5$  and 6 bits. Do this for  $\sigma_x$  varying in the range  $[-50, 0]$  dBFS and in steps of 0.1 dBFS. Plot the resulting curves (SQNR in dB vs.  $\sigma_x$  in dBFS) along with the theoretical expression

$$\text{SQNR} = 6.02N + 4.77 - 20 \log_{10} \frac{\text{FS}}{\sigma_x} \quad (\text{dB}). \quad (1)$$

Are there any differences between the theoretical and empirical curves? If so, how do you explain them?

- In view of your results, what are the optimum values (regarding SQNR) of  $\sigma_x$  (in dBFS), and for the different resolutions analyzed (3 to 6 bits)? Does this agree with your intuition (see first point above)?
- Repeat the previous points, but now using normally distributed input samples with zero mean and standard deviation  $\sigma_x$ .

■

## 3 Spectral analysis

Now we analyze the effect of quantization noise in the frequency domain by using the DFT tool, and assuming again sinusoidal input signals. Recall that the  $M$ -point DFT of the sequence  $\{x[n], n = 0, 1, \dots, M-1\}$  is given by

$$X[k] = \sum_{n=0}^{M-1} x[n] e^{-j \frac{2\pi}{M} kn}, \quad k = 0, 1, \dots, M-1,$$

and that it provides samples of the Fourier Transform<sup>4</sup>  $X(e^{j\omega})$  at  $\omega = \frac{2\pi}{M}k$ . Since  $\omega = \frac{2\pi f}{f_s}$  rad, where  $f$  is the frequency variable in Hz, and  $f_s$  is the sampling rate (also in Hz), it follows that  $X[k]$  represents the frequency content of the original analog signal at  $f = \frac{k}{M}f_s$  Hz.

<sup>2</sup>dB relative to the full-scale value, i.e.,  $20 \log_{10}(\sigma_x/\text{FS})$ .

<sup>3</sup>That is, use the command `var` to estimate the variances of the unquantized signal and of the quantization error, and then compute the ratio of the values so obtained.

<sup>4</sup>Assuming that  $x[n] = 0$  for  $n < 0$  and  $n \geq M$ .

#### □ Task 4

- Assume a full-scale sinusoidal input with  $f_0 = 37.1094$  MHz, and let the FFT size be  $M = 1024$ . Generate  $15 \cdot M$  samples of  $x(t)$  (at  $f_s = 100$  MHz) and quantize them to  $N = 12$  bits. Break the vector `xq` of quantized samples into 15 size- $M$  blocks using, e.g., the command `reshape`:

```
xqblocks = reshape(xq, M, 15);
```

so that each column of the  $M \times 15$  matrix `xqblocks` will contain the corresponding block of size  $M$ . Now, since the `fft` command computes the FFT columnwise, in order to apply an  $M$ -point FFT to each block, we simply make

```
X = fft(xqblocks, M);
```

Average the squared magnitude of the DFT coefficients over the 15 blocks and plot the results between 0 and  $f_s/2$ , in dBFS <sup>5</sup>.

Observe the location and peak value of the principal frequency component, as well as the value of the noise floor. Do your observations agree (quantitatively) with what you would expect?

- Repeat the previous steps for an FFT size  $M = 256$ .
- Set again  $M = 1024$ , and repeat the analysis for decreasing resolutions of 10, 8 and 6 bits.
- Consider again  $M = 1024$  and  $N = 12$  bits. Repeat the analysis reducing the amplitude of the sinusoid to  $1/3$  of the full scale value, and compare your observations with the theoretical prediction.
- Let  $M = 1024$ ,  $N = 12$  bits and a full-scale sinusoid. Slightly change the frequency of the sinusoid to 37.12 MHz and repeat the analysis. How do your observations change? Does it make any difference if you use a larger number of samples, say  $100 \cdot M$ ? What happens if you increase the resolution to 16 bits?

How do you explain all these?

■

---

<sup>5</sup>For this, the normalizing constant should be the squared magnitude of the peak DFT coefficient when the input is a full-scale sinusoid with frequency  $f_0 = 2\pi \frac{k_0}{M}$  for some  $k_0 \in \mathbb{N}$ .

## 4 Nonlinear distortion

Up to this point we have considered ideal quantizers. In practice, ADC components present tolerances that introduce additional distortion; for example, if the quantization stepsize is not constant over the whole input range. In that case it is common to refer to the ADC as *nonuniform*.<sup>6</sup>

We can model a nonuniform quantizer  $Q\{\cdot\}$  as the concatenation of a nonlinear mapping  $g(\cdot)$  followed by a uniform quantizer  $Q_0\{\cdot\}$ , so that  $Q\{x\} = Q_0\{g(x)\}$ . For illustration purposes, here we use a nonlinearity of the form

$$g_\gamma(x) = \text{sign}(x) \cdot \frac{\text{FS}}{\ln(1 + \gamma)} \ln \left( 1 + \gamma \frac{|x|}{\text{FS}} \right), \quad (2)$$

which is parameterized by  $\gamma$ . By using L'Hôpital's rule, you can easily check that  $\lim_{\gamma \rightarrow 0} g_\gamma(x) = x$ . Thus, for  $\gamma = 0$  the quantizer becomes uniform. As  $\gamma$  increases, one can expect the nonlinear distortion to become more pronounced. We will use spectral analysis to characterize this distortion in terms of Spurious-Free Dynamic Range (SFDR).

### □ Task 5

- Plot  $g_\gamma(x)$  vs.  $x$  in the range  $x \in [-\text{FS}, \text{FS}]$  for  $\gamma = 0, 1$  and  $2$ . For input signals whose values are always much smaller than FS (in absolute value), what will be the effect of the nonlinearity?
- Modify the code in `quant_i.m` and write a Matlab function `dquant_i.m` implementing this nonuniform quantizer. The format should be similar to that of `quant_i.m`, but including an additional input parameter `gama`:

```
xq = dquant_i( x, FS, Nbits, gama );
```

- Generate samples (at 100 MHz) of a full-scale sinusoid with  $f_0 = 6.8359$  MHz. Quantize them to  $N = 11$  bits using  $\gamma = 0.003$  in `dquant_i`. Determine the SFDR in dBFS using an FFT size  $M = 2048$ , and then with  $M = 512$ . Does the SFDR depend on the FFT size? Does the noise floor depend on the FFT size? How do you explain this?
- Using  $M = 2048$ , repeat the previous step for  $\gamma = 0.01$  and  $0.1$ . Are the spectral spurs located where you would expect?
- Set now the amplitude to  $\frac{\text{FS}}{3}$ . Using  $M = 2048$ , measure the SFDR and express it in both dBFS and dBc for  $\gamma = 0.005, 0.05$  and  $0.1$ . Will these values change if you repeat the analysis with  $M = 512$ ?
- Consider now samples (at 100 MHz and with 11-bit resolution) of a sinusoid with frequency 3.3202 MHz and amplitude  $\frac{\text{FS}}{2}$ . Obtain the THD for this nonuniform ADC with  $\gamma = 0.3$  under the IEEE 1241-2000 specification, expressed in both dB and percentage. ■

---

<sup>6</sup>Some authors refer to these ADCs as *nonlinear*, although of course all ADCs, uniform or not, are nonlinear since they must include a quantizer.

## 5 Aperture errors

One of the limiting factors of fast A/D converters is *aperture jitter*. This refers to the uncertainty in the times at which samples are obtained; we recommend reading [2] for a clarifying exposition. Basically, if  $x_c(t)$  denotes the original analog signal, the (unquantized) samples obtained by the ADC can be modeled as

$$x[n] = x_c(nT_s + \tau_n),$$

where  $T_s$  is the sampling interval ( $f_s = 1/T_s$ ), and  $\{\tau_n\}$  are random variables (having units of seconds) giving rise to aperture jitter. Usually these are modeled as uniformly distributed with zero mean and rms value  $\sigma_\tau$ , and statistically independent; typical values of  $\sigma_\tau$  are in the range of several picoseconds. Note that if  $\sigma_\tau = 0$  we obtain an ideal converter with no aperture jitter.

We can write the (unquantized) samples as

$$x[n] = x_c(nT_s) + v_\tau[n],$$

where  $v_\tau[n]$  is the noise due to aperture jitter, defined simply as  $v_\tau[n] = x_c(nT_s + \tau_n) - x_c(nT_s)$ , i.e., the difference between the *actual* and the *ideal* measurements. The Signal-to-Noise Ratio (taking only aperture jitter noise into account) is therefore

$$\text{SNR} = 10 \log_{10} \frac{E\{x_c^2(nT_s)\}}{E\{v_\tau^2[n]\}} \approx 20 \log_{10} \frac{1}{2\pi f_c \sigma_\tau} \quad (\text{dB})$$

where the last approximation applies to sinusoidal signals of frequency  $f_c$ . As you can see, the effect of aperture jitter is more pronounced for high frequency signals (why?).

### □ Task 6

Consider a uniform quantizer with 12-bit resolution and 100 MHz sampling rate.

- If the rms value of the aperture jitter is 20 ps, and the input signal is a full-scale sinusoid with frequency  $f_c$ , for which values of  $f_c$  will the aperture error power dominate the quantization noise power?
- If the rms value of the aperture jitter is 20 ps, and the input signal is a 3-MHz sinusoid, for which values of the amplitude (in dBFS) will the aperture error power dominate the quantization noise power?
- Simulate the effect of aperture jitter on a full-scale sinusoid with frequency 40.03905 MHz. Consider two cases:  $\sigma_\tau = 10$  ps and  $\sigma_\tau = 0.1$  ps respectively. Perform a 1024-FFT analysis of your data and check whether the perceived noise floor is at the expected level.
- Neglecting other possible sources of distortion, the total SNR is given by the ratio of the signal power to the sum of the powers of the noises due to jitter and quantization. Plot the theoretical

total SNR (in dB) vs. input frequency over the range 0.1–100 MHz, assuming a full-scale sinusoid and for  $\sigma_\tau \in \{10, 20, 40\}$  ps,  $N \in \{10, 14\}$  bits (so that you should have six graphs in a single plot, whose x-axis should be in log scale). Comment on your results.

■

## References

- [1] W. Kester, “Taking the mystery out of the infamous formula “SNR =  $6.02N + 1.76$  dB”, and why you should care,” Analog Devices, Application Note MT-001. Available at [www.analog.com](http://www.analog.com).
- [2] W. Kester, “Aperture Time, Aperture Jitter, Aperture Delay Time: Removing the Confusion,” Analog Devices, Application Note MT-007. Available at [www.analog.com](http://www.analog.com).