Lab 05

*Quantization*

# Quantization Process

Sampling and quantization are the necessary prerequisites for any digital signal processing operation on analog signals. A sampler and quantizer are shown in Fig. below. The hold capacitor in the sampler holds each measured sample *x(nT)* for at most *T* seconds during which time the A/D converter must convert it to a quantized sample, *x*Q*(nT)*, which is representable by a finite number of bits, say *B* bits. The *B*-bit word is then shipped over to the digital signal processor.

quantized signal *xQ*(*nT*)

*x*(*t*)

analog signal

sampler & quantizer

*x*(*nT*)

sampled signal

A/D converter

sample & hold

*T*

to DSP

*B* bits/sample

Fig. Analog to digital conversion.

After digital processing, the resulting *B*-bit word is applied to a D/A converter which converts it back to analog format generating a staircase output. In practice, the sample/hold and ADC may be separate modules or may reside on board the same chip.

The quantized sample *x*Q*(nT)*, being represented by *B* bits, can take only one of 2*B* possible values. An A/D converter is characterized by a *full-scale range R*, which is divided equally (for a uniform quantizer) into 2*B quantization levels*, as shown in

Fig. 2 The spacing between levels, called the *quantization width* or the quantizer resolution, is given by:

*Q* = 2*B*

*R*

This equation can also be written in the form:

*R/*2 3*Q*



*x*(*t*)

*x*(*nT*)

*nT*

*t*

*Q*

*xQ*(*nT*)

2*Q Q*

*R* 0

*-Q*

*-*2*Q*

*-*3*Q*

*-R/*2

quantization levels

Fig. Signal quantization.

*Q* = 2*B*

*R*

which gives the number of quantization levels. Typical values of *R* in practice are be- tween 1–10 volts. Figure shows the case of *B* = 3 or 2*B* = 8 levels, and assumes a *bipolar* ADC for which the possible quantized values lie within the symmetric range:

In Fig. quantization of *x(t)* was done by *rounding*, that is, replacing each value

*x(t)* by the value of the *nearest* quantization level. Quantization can also be done by *truncation* whereby each value is replaced by the value of the level below it. Rounding is preferred in practice because it produces a less biased quantized representation of the analog signal.

The *quantization error* is the error that results from using the quantized signal

*x*Q*(nT)* instead of the true signal *x(nT)*, that is,

*e(nT)*= *x*Q*(nT)*−*x(nT)*

In general, the error in quantizing a number *x* that lies in *[*−*R/*2*, R/*2*)* is:

where *x*Q is the quantized value. If *x* lies between two levels, it will be rounded up or down depending on which is the closest level. If *x* lies in the upper (lower) half between the two levels, it will be rounded up (down). Thus, the error *e* can only take the values

Therefore, the maximum error is in magnitude. This is an overestimate for the typical error that occurs. To obtain a more representative value for the average error, we consider the mean and mean-square values of defined by:

The result states that on the average half of the values are rounded up and half down. Thus, cannot be used as a representative error. A more typical value is the root-mean-square (rms) error defined by:

Equations can be given a probabilistic interpretation by assuming that the quantization error is a random variable which is distributed uniformly over the range that is, having probability density:

*p*(*e*)

1*/Q*

*Q/*2

0

*Q/*2

The normalization is needed to guarantee:

It follows that Eqs. represent the statistical expectations:

Thinking of and as the ranges of the signal and quantization noise, the ratio in Eq is a signal-to-noise ratio (SNR). It can be expressed in dB:

which is referred to as the 6 *dB per bit* rule.

The probabilistic interpretation of the quantization noise is very useful for deter- mining the effects of quantization as they propagate through the rest of the digital processing system.

(

*x*Q*(n)*= *x(n)*+*e(n)*

we may think of the quantized signal *x*Q*(n)* as a noisy version of the original unquantized signal *x(n)* to which a noise component *e(n)* has been added. Such an additive noise model of a quantizer is shown in Fig.



*e*(*n*)

*x*(*n*)

*x*Q(*n*)

*x*(*n*)

Quantizer

*x*Q(*n*)

# D/A Convertors

Next, we discuss some coding details for standard A/D and D/A converters, such as binary representations of quantized samples and the successive approximation method of A/D conversion. We begin with D/A converters, because they are used as the building blocks of successive approximation ADCs. We take a functional view of such converters without getting into the electrical details of their construction.

Consider a B-bit DAC with full-scale range R, as shown in Fig. Given B inputbits of zeros and ones, b [b1, b2,..., bB], the converter outputs an analog value xQ, that lies on one of the 2B quantization levels within the range R. If the converter is unipolar, the output xQ falls in the range [0, R). If it is bipolar, it falls in [−R/2, R/2).

=

*b*1 *b*2

*B* input bits *b*3

**.**

**.**

**.**

*bB*

MSB

**.**

DAC

**.**

**.**

LSB

*x*Q

analog output

*R* (reference)

Fig. 2.3.1 *B*-bit D/A converter.

The manner in which the *B* bits *[b*1*, b*2*,..., bB]* are associated with the analog value *x*Q depends on the type of converter and the coding convention used. We will discuss the three widely used types:

(a) unipolar natural binary,

(b) bipolar offset binary, and

(c) bipolar two’s complement converters.

Table summarizes the three converter types and their input/output coding conventions and Table compares the three coding conventions for the case *B* = 4 and *R* = 10 volts. The level spacing is *Q* = *R/*2*B* = 10*/*24 = 0*.*625 volts. The codes *[b*1*, b*2*, b*3*, b*4*]* in the first column, apply to both the natural and offset binary cases, but the quantized analog values that they represent are different.

|  |  |
| --- | --- |
| Converter type | I/O relationship |
| natural binary offset binary  two’s complement | *x*Q = *R(b*12−1 + *b*22−2 + ··· + *bB*2−*B)*  *x*Q = *R(b*12−1 + *b*22−2 + ··· + *bB*2−*B* − 0*.*5*) x*Q = *R(b*12−1 + *b*22−2 + ··· + *bB*2−*B* − 0*.*5*)* |

Table 2 Converter types.

For the natural binary case, the values *x*Q are positive, spanning the range *[*0*,* 10*)* volts, with the maximum value being *R* − *Q* = 10 − 0*.*625 = 9*.*375. For offset binary, the level values are offset by half scale, *R/*2 = 5 volts, and span the range *[*−5*,* 5*)* volts, with the maximum being *R/*2 − *Q* = 5 − 0*.*625 = 4*.*375 volts. Note that the upper ends of the full-scale range, *R* = 10 and *R/*2 = 5 volts, are shown in the table for reference and do not represent a level.

The last column shows the two’s complement codes. They are obtained from the first column by complementing the MSB, *b*1. The quantized values *x*Q represented by these codes are the *same* as in the offset binary case, that is, given in the fifth column of the table.

**Unipolar Natural Binary Converter**  
In the unipolar natural binary converter, the binary bits directly represent the magnitude of the analog output. The output is computed as follow:

where:  
 is the number of bits,  
 is the full-scale range: of the :

For example, if we: have a 3 bit unipolar natural binary converts with the binary code 101 , the corresponding decimal value is 5 , and the output voltage would he:

This results in an analog output voltage that less within the range (in, Rt).

Bipolar Offset Binary Converter  
In the bipolar offset binary converter, the: binary bits represent the magnitude of the: analog output, hat the converter has an offset to allow for negative values. The output is compated is:

This converter has a rang: of .

Bipolar Two's Complement Converter  
In the bipolar two's complement converter, the binary bits represent a two's complement representation of the analog value. The output is compoted as:

This converter also has a range of .

# A/D Converters

A/D converters quantize an analog value *x* so that it is represented by *B* bits *[b*1*, b*2*,..., bB]*, as shown in Fig. 2.4.1. ADCs come in many varieties, depending on how the conversion process is implemented. One of the most popular ones is the *successive approximation* A/D converter whose main building block is a D/A converter in a feedback loop. It is shown in Fig..

## x

analog input

## R

MSB

LSB

ADC

*b*1 *b*2

*b*3 *B* output bits

**.**

**.**

**.**

*bB*

Fig. *B*-bit A/D converter.

The conversion algorithm is as follows. Initially all *B* bits are cleared to zero, b = *[*0*,* 0*,...,* 0*]*, in the successive approximation register (SAR). Then, starting with the MSB *b*1, each bit is turned *on* in sequence and a test is performed to determine whether that bit should be left *on* or turned *off*. The control logic puts the correct value of that bit in the right slot in the SAR register. Then, leaving all the tested bits set at their correct values, the next bit is turned *on* in

ADC

*x*

comparator

*x*

+

-

SAR

*C* = 1/0 *b*1 *b2 b*3 *bB*

Q

*x*Q

*b*1 *b2 b*3 *bB*

DAC

# Encoding A/D Values for reconstructing signal

A straightforward method to quantize a signal involves generating a low-frequency sinusoidal waveform. This analog signal is intentionally kept at a low amplitude to simplify experimentation, allowing for easier analysis of results. By limiting the observation to just a few cycles, typically one or two, the data generated provides sufficient insight for analysis purposes.

## Preparing the Input Signal

To elaborate, this method begins by generating a sinusoidal waveform with a low frequency. The amplitude of this signal is deliberately maintained at a low level to minimize complexity. By restricting the observation period to only one or two cycles, researchers can efficiently examine the outcomes without overwhelming amounts of data. This controlled approach ensures that the experimental results remain manageable and conducive to thorough analysis.

## Conditioning the Signal

The subsequent step involves conditioning the signal to fit within a predefined range, denoted as . This range, or amplitude, is adjusted based on the characteristics of the input signal to ensure that it falls within the desired scope of observations.

To achieve this, the input signal is examined to determine its amplitude characteristics. Depending on whether the signal exceeds or falls short of the desired range appropriate scaling operations are performed to bring the signal within the specified bounds.

If the input signal's amplitude surpasses the range the signal is downscaled to fit within Conversely, if the input signal's amplitude is insufficient to cover the entire range the signal is upscaled to fully utilize the available amplitude range.

|  |  |  |
| --- | --- | --- |
|  |  | |
| Input Sample Signal |  | Scaled signal For Range R=2 |

This conditioning process ensures that the signal is tailored to suit the requirements of the observation scope, thereby facilitating accurate and meaningful analysis of the signal's behavior within the defined range

## A to D Convertor Quantizing The Signal | Encoding

The third step involves quantizing the signal using an Analog-to-Digital Converter (ADC). The role of bit depth is crucial here, as it determines the resolution of the quantized signal and hence the level of quantization error. One common method to quantize the signal is to normalize the signal's amplitude to a range between 0 and 1. And than to upscale the signal as per requirement. The steps are defined as below

1. **Normalize the Signal**

Normalizing the signal involves scaling its amplitude to a range between 0 and 1. This is achieved by subtracting the minimum value of the signal from each sample and then dividing by the range of the signal

1. **Calculate Number of Quantization Levels**

The number of quantization levels is determined by the bit depth of the ADC. For bits, there are quantization levels

1. **Quantize the Signal**

Quantizing the normalized signal involves rounding it to the nearest quantization level. This is done by multiplying the normalized signal by the number of quantization levels minus one and then dividing by. The formula for quantization is:

Based on the requirement and the necessity to reduce quantization error, function like can be used

1. **Re-Conditioning**

Reconstructing the quantized signal involves scaling it back to the original range of the signal. This is done by multiplying the quantized signal by the range of the original signal and then adding the minimum value of the original signal. The reconstructed signal ensures that the quantized signal is properly scaled and aligned with the original signal.

# Reconstructing Quantized signal

# To convert the quantized signal from digital to analog values, the zero-order hold (ZOH) . . FOH pr Ideal low pass filter can be implemented