. Digital Signal Processing — Instructor: Dr. Shafayat Abrar

. Lab 04: Quantization and Sigma-Delta Conversion — Demonstrator: Ahmad Bilal

1 Objectives

2 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. 1. The hold capacitor in the sampler holds each measured sample $x(nT_S)$ for at most T_S seconds during which time the A/D converter must convert it to a quantized sample, $x_Q(nT_S)$, 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. After digital processing, the

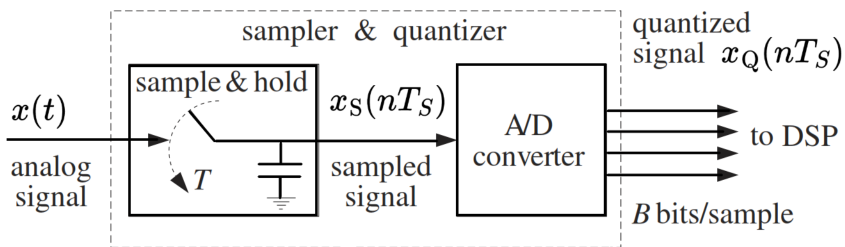

Figure 1: Sampling-holding and quantization process.

resulting B-bit word is sent to a D/A converter, which converts it back to analog format, producing a staircase output. In practice, the sample/hold and ADC may be separate modules or integrated on the same chip. The quantized sample $x_Q(nT)$, represented by B bits, can take one of 2^B possible values. An A/D converter has a full-scale range R, which is evenly divided into 2^B levels, see Fig. 2. The spacing between the levels, called the quantization width or resolution, is:

$$Q = R/2^B \tag{1}$$

Typical values of R in practice are between $1-10$ volts. Fig. 2 shows the case of $B = 3$ or $2^B = 8$ levels, and assumes a polar ADC for which the possible quantized values lie within the symmetric range:

$$-0.5R \leq x_Q(nT_S) < 0.5R \tag{2}$$

To satisfy (2), we need the following conditioning on the quantizing signal:

$$\max\{x(t)\} < \frac{R}{2}, \ \ \forall t \tag{3}$$

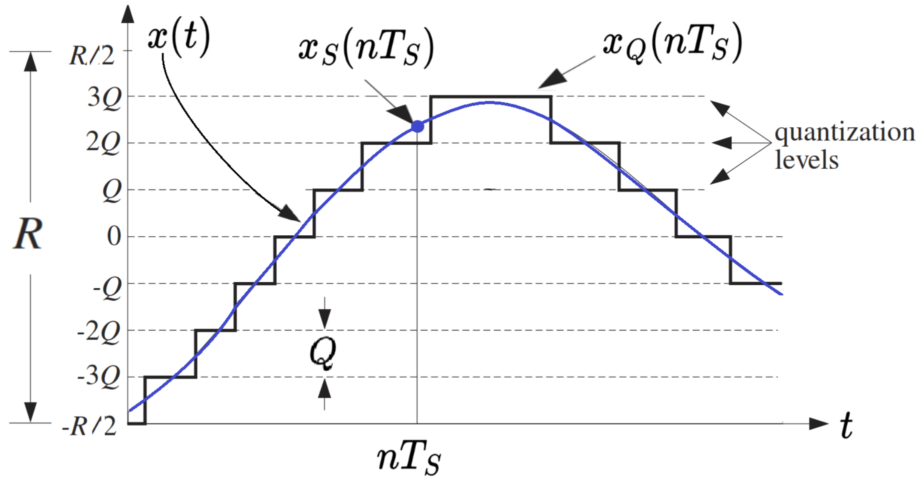

Figure 2: Signal quantization.

A typical (mid-tread) uniform quantizer with a quantization step size equal to some value Q can be expressed as

$$x_{\mathrm{Q}}(nT_S) = Q \times \left\lfloor \frac{x(nT_S)}{Q} \right\rfloor$$

where the notation $\lfloor \cdot \rfloor$ denotes the floor function.

Activity 1: Consider a sinusoidal signal $x(t) = A\cos(2\pi f_0 t)$. Assuming $B = 3$ bits, obtain a quantized copy, $x_{\mathrm{Q}}(t)$, of the quantizing signal $x(t)$. For $R = 2$, choose A such that $A < R/2$. Also, choose appropriate values for f_0 and T_S. Note use MATLAB `stairs` to plot $x_{\mathrm{Q}}(t)$

If you do this task correctly, your results must look like In Activity 1, the quantization of $x(t)$ is

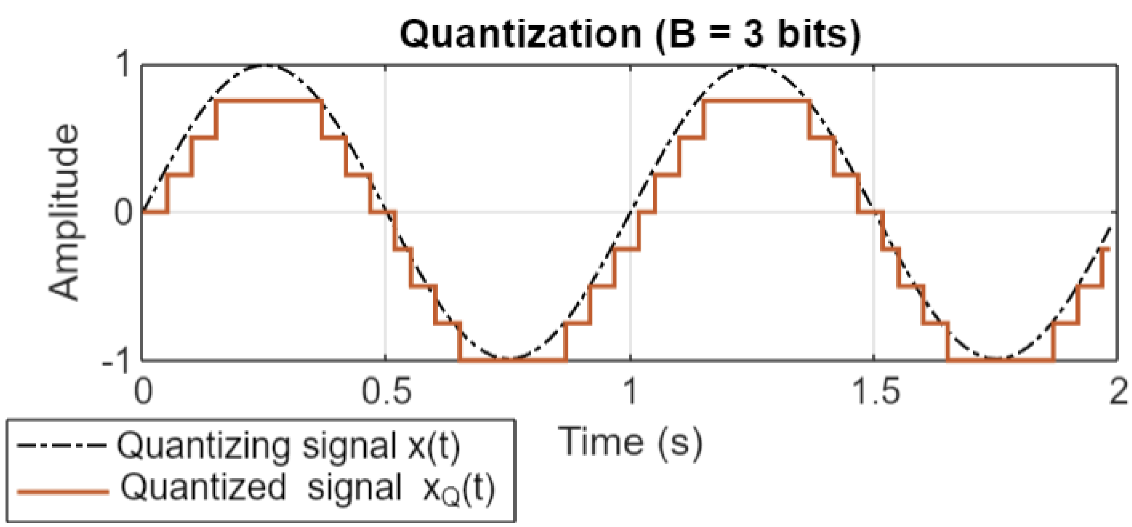

Figure 3: Caption

performed by rounding, replacing each value of $x(t)$ with the nearest quantization level. The resulting **quantization error** is the difference between the quantized signal $x_{\mathrm{Q}}(nT_S)$ and the true signal $x(nT_S)$, defined as:

$$e(nT_S) = x_Q(nT_S) - x(nT_S)$$

or equivalently:

$$e[n] = x_Q[n] - x[n]$$

Thus, the error e can only take the values

$$-\frac{Q}{2} \leq e \leq \frac{Q}{2}$$

Therefore, the maximum error is $e_{\max} = Q/2$ 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 e defined by:

$$\bar{e} = \frac{1}{Q}\int_{-Q/2}^{Q/2} e\, de = 0, \quad \text{and} \quad \overline{e^2} = \frac{1}{Q}\int_{-Q/2}^{Q/2} e^2\, de = \frac{Q^2}{12} \tag{4}$$

The result $\bar{e} = 0$ states that, on average, half of the values are rounded up and half down. Thus, $\bar{e}$ cannot be used as a representative error. A more typical value is the root-mean-square (RMS) error defined by:

$$e_{\text{RMS}} = \sqrt{\overline{e^2}} = \frac{Q}{\sqrt{12}}$$

Activity 2A: For the signal considered in Activity 1, obtain the plot of quantization error, $e(t)$, and show that the mean and variance of $e(t)$ satisfy or close to Equation (4).

Activity 2B: Discuss the effect of increasing sampling frequency on the RMS value of quantization error for the fixed value of quantization bits B.

Activity 2C: Discuss the effect of increasing quantization bits B on the RMS value of quantization error for the fixed value of sampling frequency.

3 Assigning Digital Bits to Quantized Levels: Case $R = 2$

Analog-to-Digital Converters (ADCs) quantize an analog signal x, encoding it into B bits $[b_1, b_2, \ldots, b_B]$. **It is assumed that $R = 2$; this ensures that all quantized levels, whether positive or negative, have amplitudes less than one.** The Two's complement method is used for bit assignment, where the most significant bit (MSB) of a negative number is always set to 1, i.e., the bit $b_1 = 1$ for the negative number, and the bit $b_1 = 0$ for the positive number.

The smallest negative number that can be represented is: -1

The largest positive number that can be represented is: $1 - 2^{1-B}$

$B = 3$ bits	
x_{Q}	$b_1b_2b_3$
0.75	011
0.50	010
0.25	001
0.000	000
-0.25	111
-0.50	110
-0.75	101
-1.00	100

$B = 4$ bits	
x_{Q}	$b_1b_2b_3b_4$
0.875	0111
0.750	0110
0.625	0101
0.500	0100
0.375	0011
0.250	0010
0.125	0001
0.000	0000
-0.125	1111
-0.250	1110
-0.375	1101
-0.500	1100
-0.625	1011
-0.750	1010
-0.875	1001
-1.000	1000

Figure 4: Caption

3.1 Converting a Two's Complement into the Quantized Value: Case $R = 2$

1. In Two's complement format, the most significant bit, MSB, (the left most bit) is 0 for positive numbers and it is 1 for negative numbers.

2. If the MSB is 0, the quantized amplitude is positive, and the decimal number is obtained as (this is true for $R = 2$ only)

$$x_Q = 2\Big(b_2 \times 2^{-2} + b_3 \times 2^{-3} + \cdots + b_B \times 2^{-B}\Big)$$

Example: Consider the bit pattern 0011 (that is B is 4). Since the MSB is 0, this is a positive number. The quantized level is obtained as

$$x_Q = 2\Big(0 \times 2^{-2} + 1 \times 2^{-3} + 1 \times 2^{-4}\Big) = 2^{-2} + 2^{-3} = 0.25 + 0.125 = 0.375$$

3. If the MSB is 1, the quantized amplitude is negative, the bit pattern $b_2b_3 \cdots b_B$ are first complemented to get $\bar{b}_2\bar{b}_3 \cdots \bar{b}_B$, then add 1 in the bit format to obtain a new bit format $\widehat{b}\widehat{b}_2\widehat{b}_3 \cdot \widehat{b}_B$, and the decimal number is obtained as (this is true for $R = 2$ only)

$$x_Q = -2\Big(\widehat{b}_2 \times 2^{-2} + \widehat{b}_3 \times 2^{-3} + \cdots + \widehat{b}_B \times 2^{-B}\Big)$$

Example: Consider the bit pattern 1011 (that is B is 4). Since the MSB is 1, this is a negative number. The bit pattern $\bar{b}_2\bar{b}_3 \cdots \bar{b}_B$ is complemented and added 1,

$$\overline{011} + 1 = 100 + 1 = 101 = \widehat{b}_2\widehat{b}_3\widehat{b}_4$$

The x_Q is obtained as

$$x_Q = -2\Big(1 \times 2^{-2} + 0 \times 2^{-3} + 1 \times 2^{-4}\Big) = -2\Big(0.25 + 0 + 0.0625\Big) = -0.625$$

Activity 3: Write a MATLAB code to convert a two's-complement binary number into a decimal number assuming that $R = 2$, i.e., the magnitude of all decimal numbers are less unity.

3.2 Converting a Quantized Value into Two's Complement Number: Case $R = 2$

Consider the MATLAB code below, which converts a given real number $-1 \leq x < 1$ into a B-bit two's complement representation.

```
function bitpattern= x2tscomp(xQ,B)
realNumber = abs(xQ);
for i = 2:B
    realNumber = 2*realNumber;
    if xQ<0
        if realNumber>1
            realNumber = realNumber-1;
            b(i) = 1;
        else
            b(i) = 0;
        end
    else
        if realNumber >= 1
            realNumber = realNumber-1;
            b(i) = 1;
        else
            b(i) = 0;
        end
    end
end
if xQ >= 0
    bitpattern = b(:)';
else
    bitpattern = not(b(:))'; %
    bitpattern(1) = 1;
end
end
```

Activity 4: Update the MATLAB code `x2tscomp(xQ,B)` to enable it to convert a vector of data, `xQ`, into two's complement format, allowing it to generate the tables shown in Fig. 4 in a single execution.

4 Representation of Two's Complement Numbers with Integer and Fractional Parts: Case $R > 2$

Representing a number with both integer and fractional parts in two's complement format is explained as follows:

1. **Define the Bit Allocation:**
   - Choose the total number of bits B, and decide how many bits will represent the integer part (N_{int}) and how many bits will represent the fractional part ($N_{\text{frac}} = B - N_{\text{int}}$). Note that the most significant bit of the integer part is used to denote the sign of the number, and the remaining $N_{\text{int}} - 1$ are used to represent magnitude.
2. **Range of Representable Values:**
   - For B-bit numbers:

$$\begin{aligned} \text{Smallest value (negative)} &: -2^{N_{\text{int}}-1}, \\ \text{Largest value (positive)} &: 2^{N_{\text{int}}-1} - 2^{-N_{\text{frac}}}. \end{aligned}$$

   - This can be verified. Previously, we have considered $N_{\text{int}} = 1$ (for sign bit) and $N_{\text{frac}} = B - 1$. The smallest value (negative) $= -2^{N_{\text{int}}-1} = -2^{1-1} = 2^0 = -1$. Similarly, the largest value (positive) $= 2^{N_{\text{int}}-1} - 2^{-N_{\text{frac}}} = 2^{1-1} - 2^{-(B-1)} = 1 - 2^{1-B}$.
3. **Convert the Number to Binary:**
   - **Integer Part:**
     - Convert the integer part of the decimal number to binary (e.g., for -3.625, the integer part is -3).
     - For negative integers, use the two's complement representation of the integer.
   - **Fractional Part:**
     - Convert the fractional part (e.g., 0.625) into binary by multiplying it repeatedly by 2. Record the integer parts of the results as binary digits until N_{frac} bits are reached or the fractional part becomes zero. This is explained in Sec 3.2.
4. **Combine Integer and Fractional Parts:**
   - Concatenate the binary representation of the integer and fractional parts into a single B-bit string.
   - Ensure the most significant bit (MSB) is the sign bit for two's complement.
5. **Pad or Truncate:**
   - If the result has fewer than B bits, pad with zeros or truncate appropriately.

Example: Representing -3.625 in Two's Complement with $N_{\text{int}} = 4$ and $N_{\text{frac}} = 4$:

Step 1: Convert Integer Part

$$\begin{aligned}&\text{Integer part} = -3,\\&\text{Binary (4 bits)} : 1101 \,\text{(two's complement for } -3\text{)}.\end{aligned}$$

Step 2: Convert Fractional Part

$$\begin{aligned}&\text{Fractional part} = 0.625,\\&0.625 \times 2 = 1.25 && \rightarrow \text{record } 1,\\&0.25 \times 2 = 0.5 && \rightarrow \text{record } 0,\\&0.5 \times 2 = 1.0 && \rightarrow \text{record } 1,\\&0.0 && \rightarrow \text{terminate (no remainder)}.\\&\text{Binary (4 bits)} : 1010.\end{aligned}$$

Step 3: Combine

$$\text{Combined binary} : 1101.1010.$$

Step 4: Adjust for Total Bits

$$\text{Final representation} : 11011010 \,\text{(8 bits, two's complement for } -3.625\text{)}.$$

Activity 5: Develop the MATLAB code to represent Two's complement numbers with integer and fractional parts where the input is a real number with a non-zero integer part.