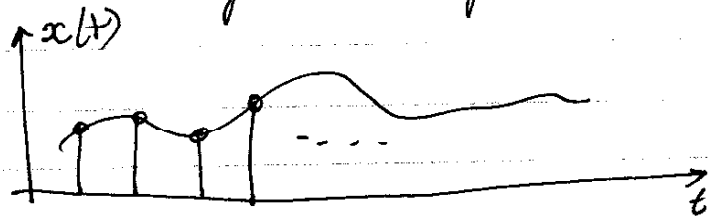


Lec 27, Mon 10/29/01

①

1.7.11 Quantization.

Suppose we have a continuous-time, real-valued signal. How do we store and manipulate it on a digital computer?



Well, one thing we need to do is to retain just a discrete set of ~~samples~~ ^{values}, because we cannot store an uncountable set of values. We considered a number of possibilities here. The simplest one was to retain several samples of the signal. More generally, we could decompose the signal in a basis and ~~store~~ ^{store} several coefficients:

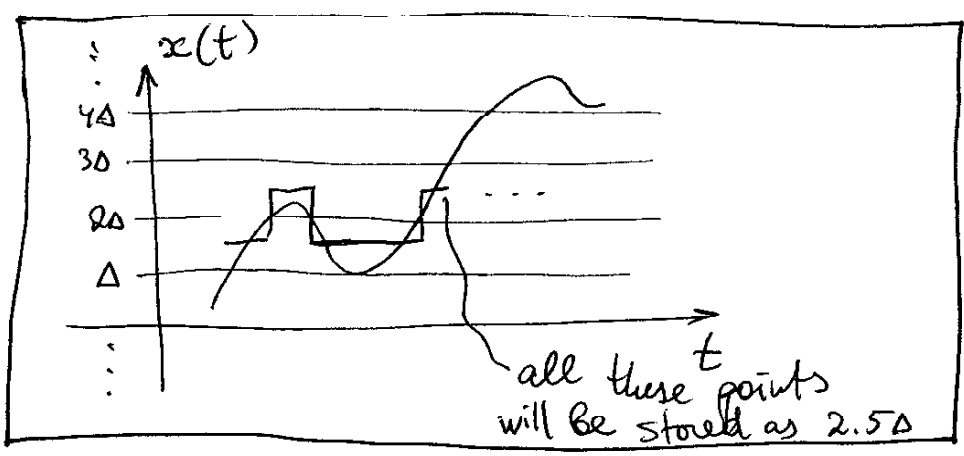
$$x(t) = \sum_{k=-\infty}^{\infty} a_k g_k(t) ; \text{ store } a_k\text{'s}$$

The ~~usual~~ We saw that the usual sampling is just a particular case of this, when g_k 's are sine functions.

However, we are not done yet. If we have a ~~discrete set of~~ finite set of ~~numbers~~ ^{samples}, each of which can take on any real value, we still would not be able to store our signal on a computer. What we need to do now is to quantize the vertical axis:

~~2/2/20~~

②



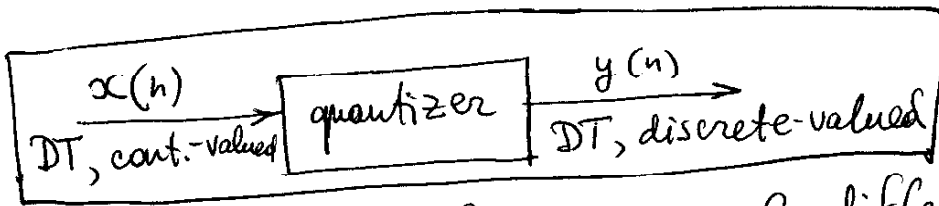
E.g., Δ = precision of the computer.

But: why bother storing pixel values of a grayscale image as 32-bit floating point numbers if 256 levels are usually enough for high visual quality?

Images: 32-bit floating pt "redundant"
 8-bit unsigned int usually enough

Terminology:
 each pixel can assume 256 different values: "8 bits per pixel"
 " " " " 16 " " : "4 " " " "

③

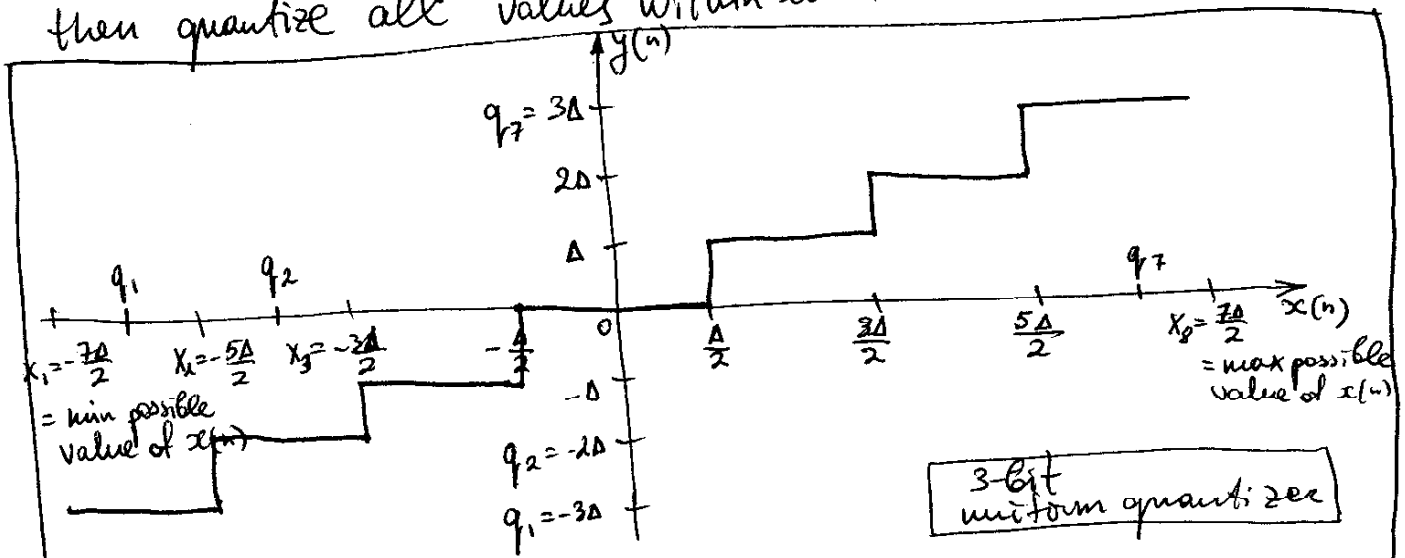


The resulting signal is, in general, different from the input signal. We would like to make this difference small, in some sense

error (distortion): $y(n) - x(n)$

Uniform Quantizer (Lab 8, Sect. 3.3.4)

Partition the range of $x(n)$ into several equal bins, and then quantize all values within each bin to the middle value



Notation:

$q_1, \dots, q_N =$ quantization levels

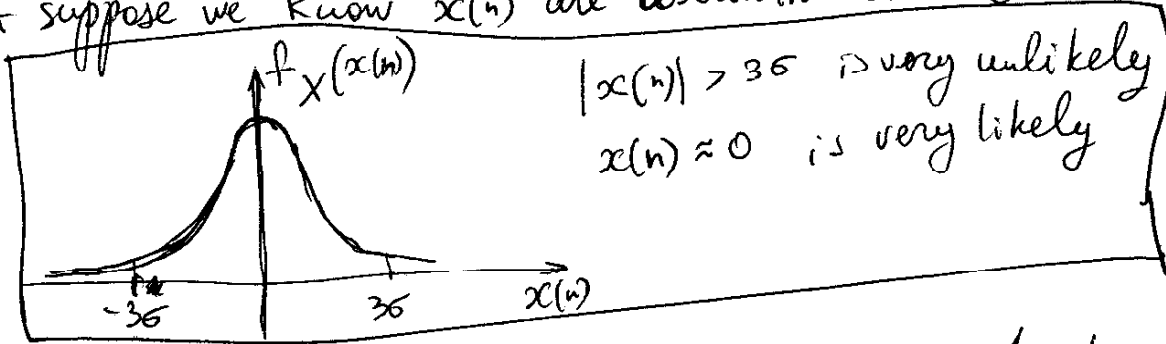
$[x_1, x_2), [x_2, x_3), \dots, [x_N, x_{N+1}) =$ corresp. quantization intervals

$\min(x(n))$
(could be $-\infty$)

$\max(x(n))$
(could be $+\infty$)

Is this a good strategy? Yes, if we do not know much about the distribution of values of $x(n)$ between $\min(x(n))$ and $\max(x(n))$.

But suppose we know $x(n)$ are observations of a Gaussian r.v.!

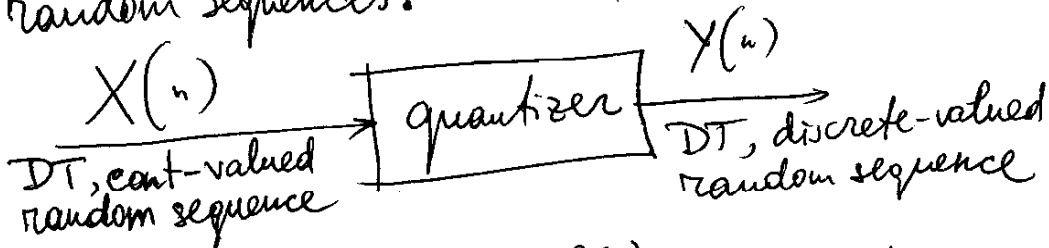


Then, e.g., it does not make sense to waste two separate quantization levels on 10005 and 10015 since we're unlikely to ever see either of these values. The correct strategy in this situation is:

- use more bits to represent values near 0;
 - use fewer bits for large values
- ⇒ should reduce mean square error.

Max Quantizer. (Lab 8, Sect. 3.5)

Model the input and output of the quantizer as random sequences:



Suppose the pdf of $X(n)$ is $f(x)$, and that

- $f(x)$ does not contain δ 's
- $f(x) > 0$ for $-\infty < x < \infty$.

Objective: find $x_1, \dots, x_{N+1}, q_1, \dots, q_N$ so as to minimize the mean-square error:

$$\begin{aligned} \text{minimize } E[(Y(n) - X(n))^2] &= \int_{-\infty}^{\infty} (y - x)^2 f(x) dx \\ &= \sum_{k=1}^N \int_{x_k}^{x_{k+1}} (y - x)^2 f(x) dx \\ &= \sum_{k=1}^N \int_{x_k}^{x_{k+1}} (q_k - x)^2 f(x) dx \end{aligned}$$

\uparrow original value
 \uparrow discretized value

Minimize w.r.t. q_k 's:

$$\frac{\partial E[(Y(n) - X(n))^2]}{\partial q_k} = 0$$

(5)

$$\int_{x_k}^{x_{k+1}} 2(q_k - x) f(x) dx = 0$$

$$\int_{x_k}^{x_{k+1}} q_k f(x) dx = \int_{x_k}^{x_{k+1}} x f(x) dx$$

$$q_k = \frac{\int_{x_k}^{x_{k+1}} x f(x) dx}{\int_{x_k}^{x_{k+1}} f(x) dx} = E[X(n) | x_k \leq X(n) \leq x_{k+1}]$$

Therefore, the k -th quantization level is the conditional expectation of $X(n)$ given that it falls within the k -th quantization interval.

Now minimize w.r.t. x_k 's:

• Since, by assumption, $f(x) \neq 0$, $x_1 = -\infty$ and $x_{N+1} = \infty$.

• For $2 \leq k \leq N$,

$$\frac{\partial E[(Y(n) - X(n))^2]}{\partial x_k} = 0$$

$$\frac{\partial}{\partial x_k} \left\{ \int_{x_{k-1}}^{x_k} (q_{k-1} - x)^2 f(x) dx + \int_{x_k}^{x_{k+1}} (q_k - x)^2 f(x) dx \right\} = 0$$

$$(q_{k-1} - x_k)^2 f(x_k) - (q_k - x_k)^2 f(x_k) = 0$$

$$(q_{k-1} - q_k) 2 \left\{ \frac{q_{k-1} + q_k}{2} - x_k \right\} f(x_k) = 0$$

$$x_k = \frac{q_{k+1} + q_k}{2}$$

I.e., we need to solve a nonlinear system of $2N-1$ equations:

$$\left\{ q_k = \frac{\int_{x_k}^{x_{k+1}} x f(x) dx}{\int_{x_k}^{x_{k+1}} f(x) dx}, \quad k=1, 2, \dots, N \right. \quad (1)$$

$$\left. x_k = \frac{q_{k-1} + q_k}{2}, \quad k=2, \dots, N \right\} \quad (2)$$

(5)

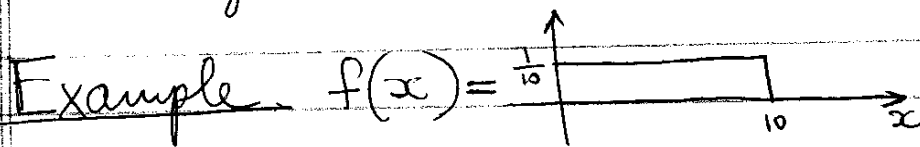
Remarks

1. Can find a closed-form solution only for very simple f 's. In general, find an approximate solution numerically, via an iterative algorithm (Lloyd's command in MATLAB):

- guess q_k 's
- find x_k 's from (2)
- find a better guess for q_k 's from (1)
- ~~find~~ re-calculate x_k 's from (2)
- etc.

2. The argument above is easily modified to accommodate $f(x)$ containing δ -funs and $f(x) = 0$ for some x 's.

3. If $f(x)$ is unavailable, ~~use~~ estimate it, using, e.g., a histogram.



Then $x_1 = 0, x_{N+1} = 10,$

$$\text{and } q_k = \frac{\int_{x_k}^{x_{k+1}} x \cdot \frac{1}{10} dx}{\int_{x_k}^{x_{k+1}} \frac{1}{10} dx} = \frac{\frac{x^2}{2} \Big|_{x_k}^{x_{k+1}}}{x \Big|_{x_k}^{x_{k+1}}} = \frac{1}{2} \frac{x_{k+1}^2 - x_k^2}{x_{k+1} - x_k} = \frac{x_k + x_{k+1}}{2}$$

\Rightarrow uniform quantizer is Max quantizer in this case (last question on Lab 8).