QUANTIZZAZIONE

In generale, sappiamo che nel processo di *campionamento*, si esegue il passaggio:

$$x(t) \longrightarrow x(nT) \ \in \mathbf{R}$$

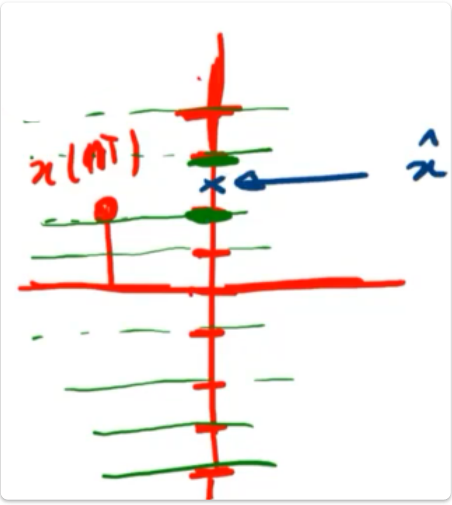
Dato che "ogni campione è un numero reale", ha bisogno di un numero infinito di cifre per essere memorizzato. Pertanto si cerca di **approssimare** questi valori con delle quantità appartenenti a un **alfabeto discreto**. Si cerca cioè di approssimare ogni campione con uno dei possibili livelli (il più vicino) descritti dal range di valori che ho a disposizione nell'alfabeto discreto.
Il valore che subisce questa alterazione viene denominato **quantizzato**, e quindi il processo diventa il seguente:

$$x(t) \longrightarrow \underbrace{x(nT)}_{\in \mathbf{R}} \longrightarrow \underbrace{\hat{x}(nT)}_{\substack{\#\text{ finito di cifre} \\ \text{(cifre binarie)}}}$$

L'errore che si commette quantizzando un campione è **irreversibile**. Non si può infatti "tornare indietro" all'esatto valore di partenza.
Matematicamente si può esprimere in questo modo:

$$\mathrm{e}(nt) = \hat{x}(nt) - x(nt)$$

- Viene per questo chiamata **rappresentazione Lossy** ovvero *con perdita*.
- L'errore commesso è modellabile (e rappresentabile) in generale come una **variabile aleatoria**



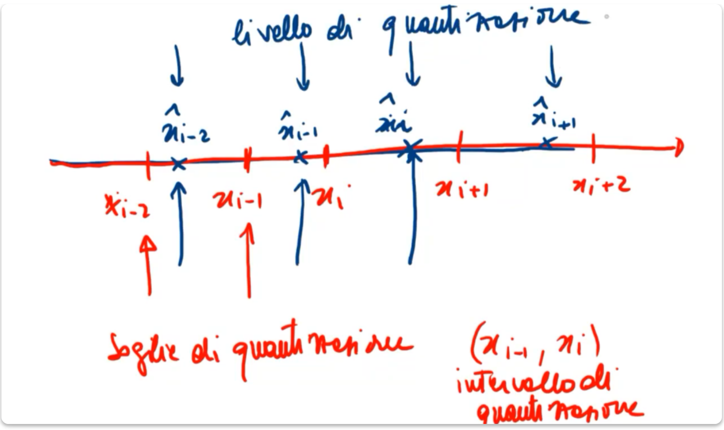
Si passa quindi da un valore $x(nT)$ a $\hat{x}(nT)$ **valore quantizzato**, ovvero:

$$x(nT) \longrightarrow \hat{x}(nT)$$

> > ***Nota: NON*** *si può effettuare il passaggio inverso, perché non c'è una correlazione $1:1$ ma è piuttosto una correlazione $\text{molti}:1$, perciò*

$$x(nT) \not\longleftarrow \hat{x}(nT)$$

La quantizzazione viene quindi effettuata utilizzando **degli intervalli e delle soglie di quantizzazione** (in rosso), e all'interno di ognuno selezioniamo un unico **livello di quantizzazione** (in blu) che sarà il riferimento associativo per tutti i campioni che cadono all'interno del relativo intervallo.



Quantizzare un segnale significa quindi eseguire il passaggio:

$$\boxed{x(nT) \xrightarrow{\text{individuare}} \boxed{\text{intervallo quantizzazione}} \xrightarrow{\text{associare}} \hat{x}_i \quad con \quad e(nT)}$$

Da notare che come detto abbiamo **un solo** $\hat{x}(t)$ per ogni intervallo di quantizzazione, perciò è importante scegliere le soglie e i livelli di quantizzazione per minimizzare l'errore a seconda del segnale d'ingresso.

- Un algoritmo che ci aiuta in questi casi (per ridurre l'errore) è l'**algoritmo di Max-Lloyd** che produce livelli e soglie ottime in relazione alla densità di probabilità dei campioni t.c. la potenza dell'errore sia minima:

$$p_{x(nT)}(x) \rightarrow x_i,\ \hat{x}_i \quad \text{t.c. } E[e^{2(nT)}] \text{ sia minima}$$

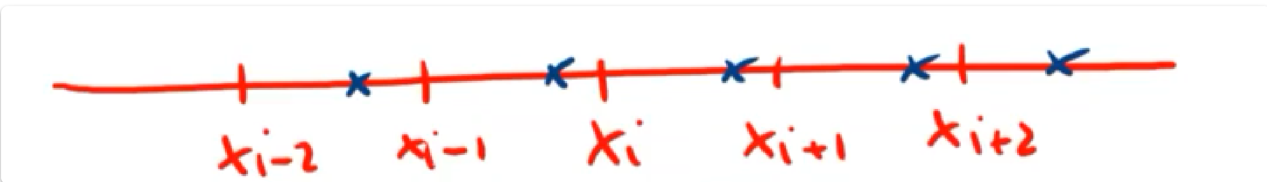
- Nota negativa: non facile calcolare intervalli e soglie ottime

QUANTIZZATORE UNIFORME

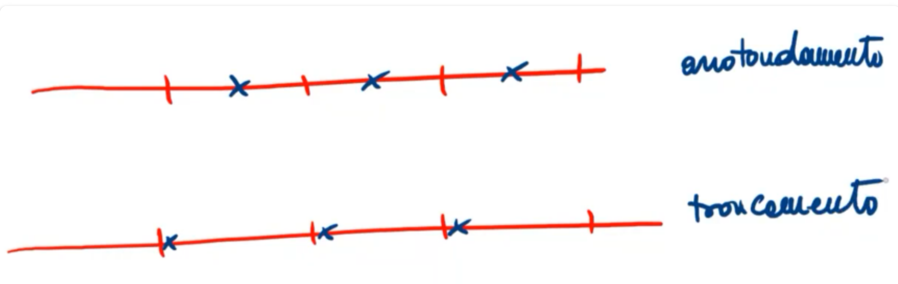
È un quantizzatore ben più semplice rispetto a quanto descritto dall'algoritmo di Max-Lloyd

- Si scelgono **soglie e livelli di quantizzazione equispaziati**
  - Pertanto,

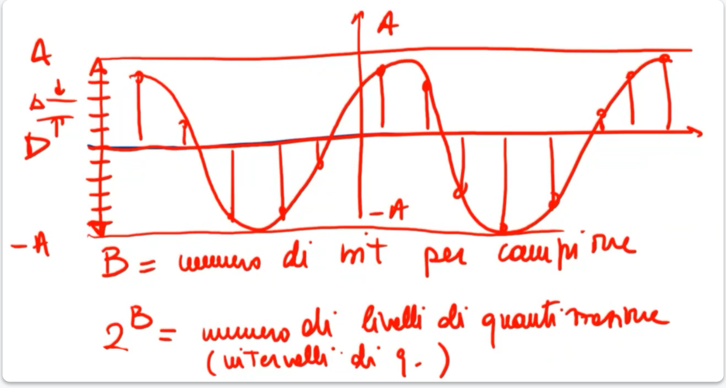
$$\begin{cases} x_i - x_{i-1} = \Delta \\ \hat{x}_i - \hat{x}_{i-1} = \Delta \end{cases} \qquad \text{con } \Delta = \text{costante} = \text{passo di quantizzazione}$$



(generalmente in realtà sceglieremo i livelli di quantizzazione al centro dell'intervallo (*arrotondamento*) oppure coincidenti con l'estremo sinistro (*troncamento*)



Segnale Sinusoidale



Supponendo la coincidenza [Dinamica - Passo di quantizzazione]:

$$\Delta = \frac{D}{2B}$$