DIGITALIZZAZIONE DEI SEGNALI

Per la prima parte di audio vedere lezione ME-8

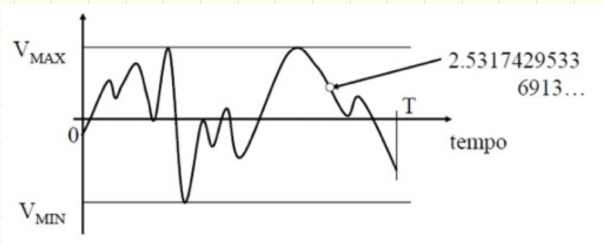
DIGITALIZZAZIONE ⟶ CAMPIONAMENTO ⟶ QUANTIZZAZIONE ⟶ CODIFICA

I computer lavorano su grandezze **digitali**. Ma ovviamente i computer devono interfacciarsi con un mondo che è invece **analogico**.
Di conseguenza abbiamo bisogno di **due convertitori**: uno da analogico a digitale ed uno da digitale ad analogico; questi due componenti sono i **bottleneck** dell'intero sistema.

Ogni dispositivo elettronico che si interfaccia con il mondo reale ha diversi convertitori. Anche un semplicissimo vecchio telefono cellulare degli anni 2000 aveva decine di convertitori ad e da.

COME RAPPRESENTARE UNA GRANDEZZA ANALOGICO TRAMITE CIFRE

Una degli esempi più semplici da comprendere è il **termometro al mercurio**; il livello dl liquido varia in modo **continuo**, ma la scala riportata al lato è **discreta**. Facciamo quindi un salto da una grandezza continua ad una discreta. Questo è il primo esempio di **quantizzazione di una grandezza**

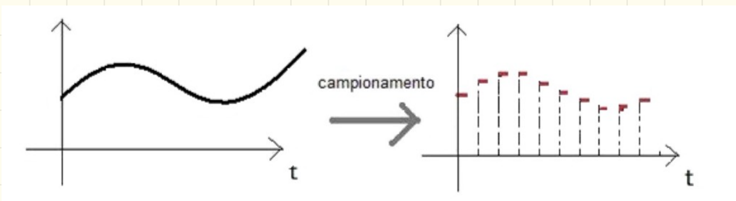


In segnale analogico è compreso tra un valore **minimo** ed un valore **massimo**. Se andiamo poi a vedere quanto vale il segnale in un qualsiasi punto x0, avremo un **numero infinito di cifre,** ovvero abbiamo una **risoluzione infinita**.

Come spostiamo il segnale analogico in digitale?
Dobbiamo effettuare due operazioni:
- **Asse delle x** (tempo): dobbiamo avere un numero finito di campioni; usiamo il teorema del campionamento.
- **Asse delle ampiezze:** il anche in questo caso dobbiamo avere un **numero finito di livelli possibili**.

Questa operazione è nota come **doppia discretizzazione**.

CAMPIONAMENTO

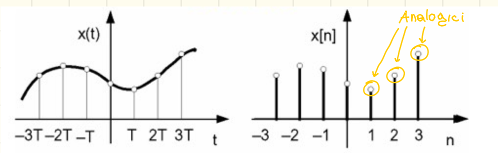


Nell'operazione di campionamento consiste nel **prelevare ad intervalli regolari il valore della grandezza analogica.**

Un segnale campionato è una sequenza di valori analogici.

Se l'intervallo di campionamento è troppo grande, non riusciamo ad avere un segnale campionato in maniera corretta; non possiamo avere una successione che descrive in maniera accurata il segnale analogico.

Il problema di questa operazione è che **i campioni sono ancora analogici**.

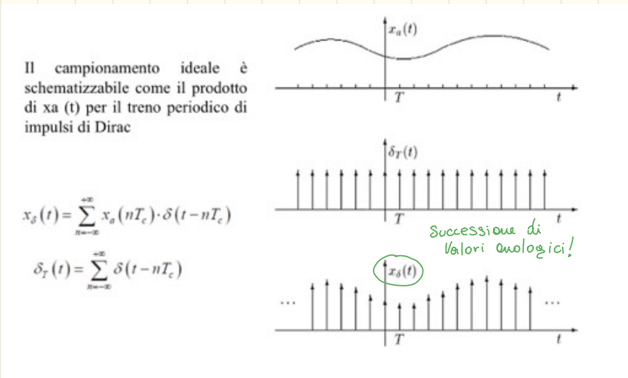


Abbiamo quindi una **discretizzazione dell'asse del tempo** (ma non dell'asse delle ampiezze). La **frequenza di campionamento** è data dal **periodo di campionamento**:

$$t = n \cdot T \Rightarrow f_c = \frac{1}{T}$$

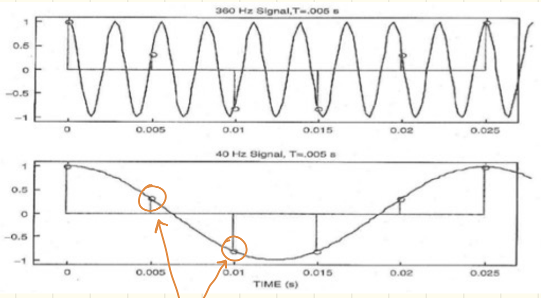
numero di campioni

Tempo tra due campioni



Con l'operazione di campionamento discretizziamo si l'asse dei tempi, ma **i valori delle ampiezze rimangono analogici**! Di conseguenza non abbiamo ancora finito.

Quanti campioni (al secondo) devo prendere per avere un buon risultato?

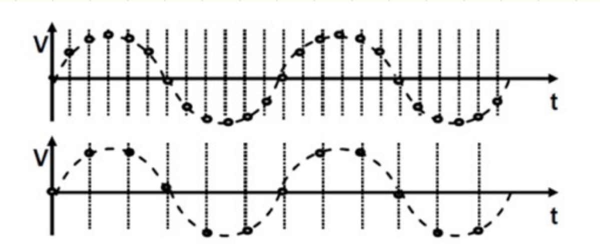


Un solo punto per ogni periodo => MALE

Con questo esempio possiamo vedere come **prendendo un intervallo di campionamento troppo alto**, il segnale campionato ottenuto sarà **molto diverso** da quello che dovevamo campionare.

Per avere un buon campionamento. **Se il segnale ha una certa banda B la frequenza di campionamento deve essere il doppio della banda del segnale.**

Es:
Se ho un segnale di 50Hz, la frequenza di campionamento deve essere almeno 100Hz, ovvero per ogni periodo devo prendere (campionare) almeno due punti.



Entrambe OK, ma diverso dettaglio

In questi due esempi stiamo campionando correttamente il segnale.

2 Pt a periodo nelle sinusoidi

$$\boxed{f_c \geq 2 f_{S\,MAX}}$$

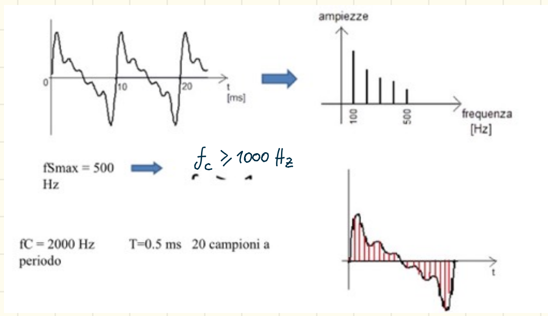
Freq di Campionamento

Freq Max del segnale da campionare $f_{SMAX} = B_S$

MONDO REALE...

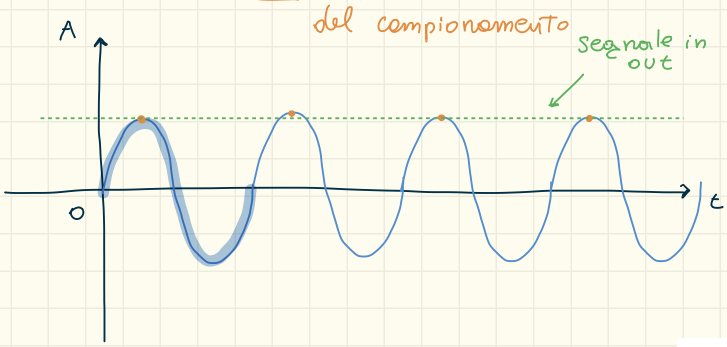
- Quello che abbiamo detto è valido se **la banda del segnale da campionare è limitata**. Nel mondo reale questo segnale non esiste. Generalmente rendiamo i segnali a banda limitata tramite dei **filtri passa-basso** in ingresso, così da avere un segnale a banda limitata.

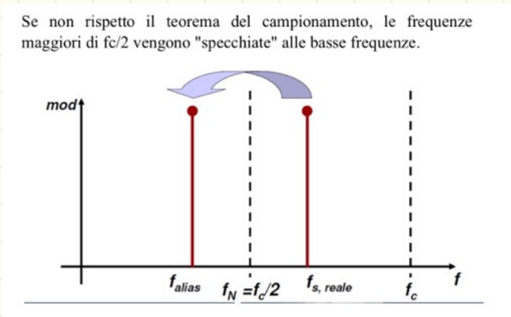
- Il segnale deve essere campionato in modo sincrono con i massimi ed i minimi del segnale da campionare. Se abbiamo quindi un transitorio, dobbiamo campionarlo ad una frequenza elevata in maniera da poterlo rappresentare.



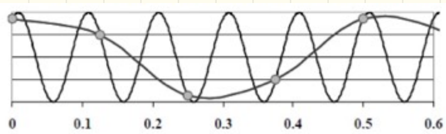
ALIASING

NON rispetto il teorema del campionamento





Se la frequenza di campionamento è la stessa frequenza del segnale da campionare, il segnale risultante sarà un segnale **costante**, ovvero a frequenza nulla!

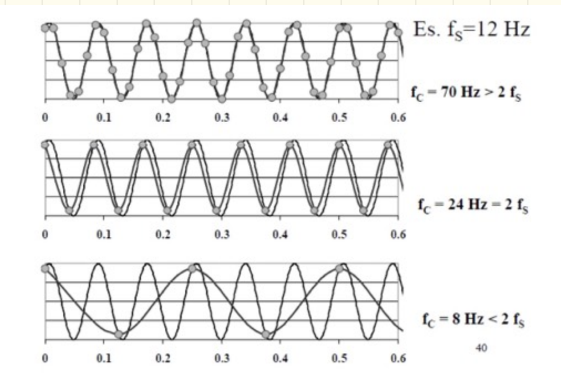


Es:
$f_S = 10$ Hz
$f_C = 8$ Hz

Δt camp. $= 1/f_C = 0.125$ s
Periodo apparente 0.5 s $\Rightarrow$ Freq apparente 2 Hz

Se $f_C < 2 f_S$ allora l' "aliasing" si manifesta

39



BENISSIMO

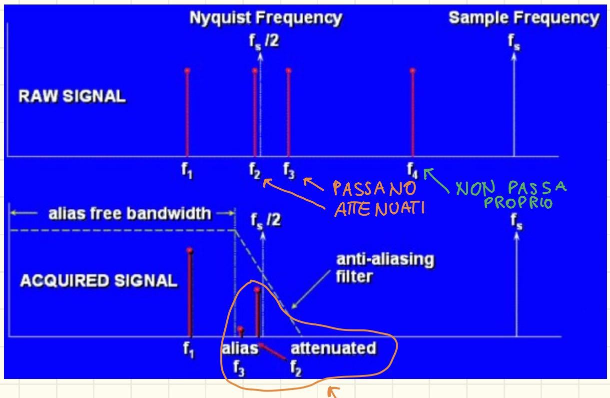
GIUSTO GIUSTO...

MALE! ⇓ ALIASING

FILTRI ANTI ALIASING

Non possiamo correggere l'aliasing una volta che si è verificato.

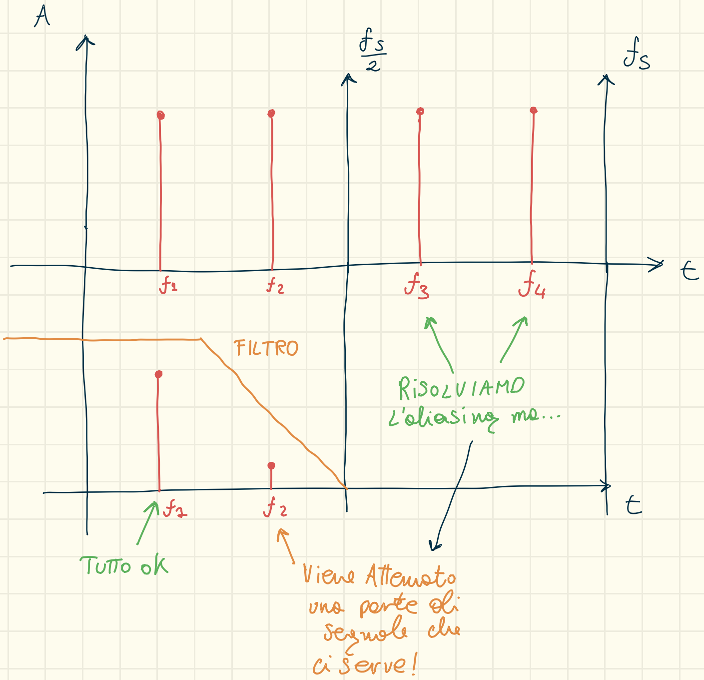
Quello che possiamo fare è **evitarlo**: possiamo ovviamente **alzare la frequenza di campionamento**, possiamo anche aggiungere un **filtro anti-aliasing** prima del convertitore a/d; questo è un filtro passa basso: la frequenza dalla quale si filtra il segnale è proprio fc/2. In questo modo limitiamo le frequenze maggiori di fc/2 ed evitiamo il fenomeno di aliasing.



F_2 ed F_3 vengono attenuati
=> F_3 lo stesso compare come ALIAS

Il problema dei filtri anti-aliasing

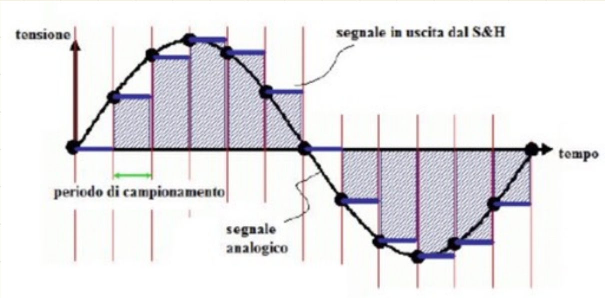
Abbiamo visto che frequenze come quella di f3 passano lo stesso seppur attenuate, e vadano a creare aliasing. Possiamo quindi pensare di **spostare il filtro** facendo si che la zona di transizione del filtro si esaurisca proprio ad Fs/2. Questa soluzione risolverebbe l'aliasing, ma una parte dei segnali tra 0 ed Fs/2 (che vogliamo che passino interamente) vengono attenuati!



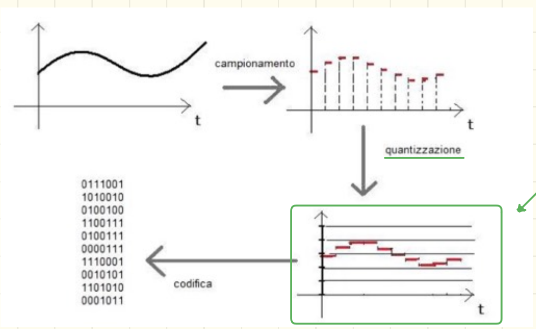
#Domande esame

Possibile domande esame sull'aliasing: quando avviene e come si risolve

QUANTIZZAZIONE



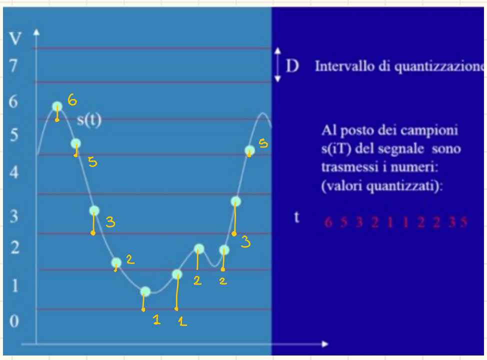
Ricordiamo che il sample & hold mantiene costante il segnale analogico per un determinato lasso di tempo. Il convertitore a/d deve convertire poi questo valore da analogico a digitale.

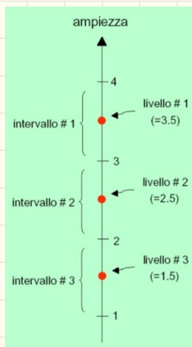


Con il processo di **quantizzazione** passiamo da tanti numeri continui (precisione infinita) a dei valori discreti (finiti).

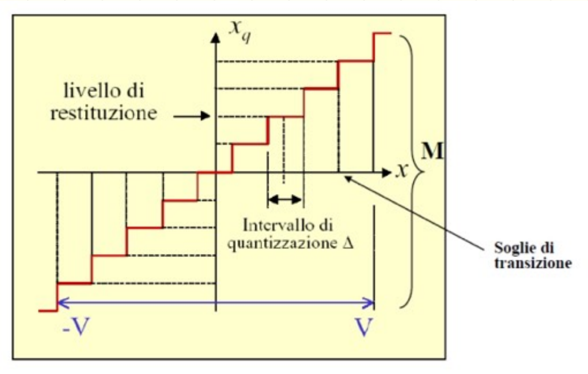
Come accade con il termometro al mercurio, il processo di quantizzazione consiste nel **confrontare i valori analogici con una scala di valori digitali (discreti)**, con l'accortezza di **leggere il valore <u>inferiore</u> più vicino a quello analogico**.

Abbiamo quindi diversi **intervalli di quantizzazione.** Se abbiamo un valore analogico compreso tra 3 e quattro, il valore quantizzato è 3.

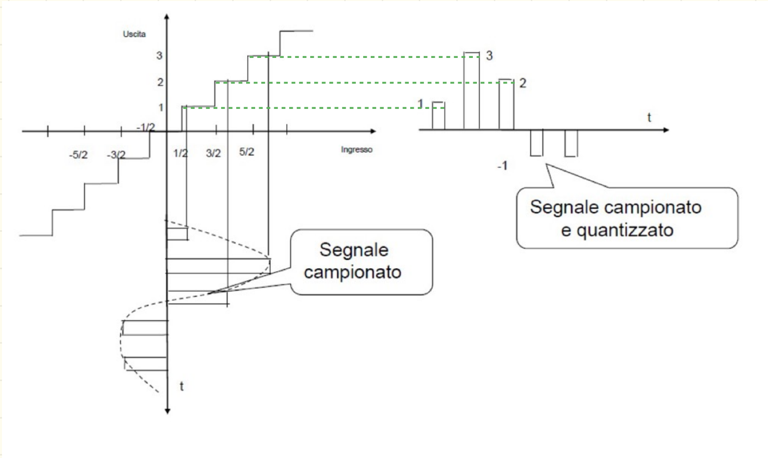




FDT DEL QUANTIZZATORE



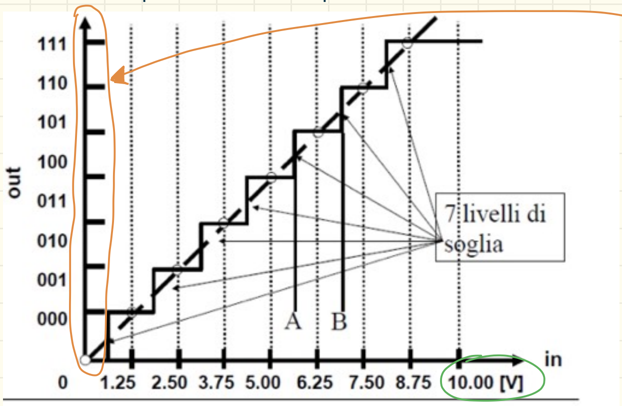
{ Asse x → INPUT (ANALOG)
{ Asse y → OUTPUT (QUANTIZZATO)



RICAPITOLANDO

Segnale Analog
↓
Operazioni con filtro Anti-Aliasing
↓
Sequenza di valori analogici
↓
quantizzazione dei valori analog ← Ancora non si parla di convertitore A/D

Ovviamente possiamo far corrispondere un certo numero binario per ogni livello di quantizzazione.



FONDO SCALA
FS = 10.00V - 0.00V = 10.00V

$$Q = \frac{FS}{2^b}$$

FONDO SCALA ovvero il valore max del segnale analogico (INPUT)

Quanto, ovvero la RISOLUZIONE del Convertitore

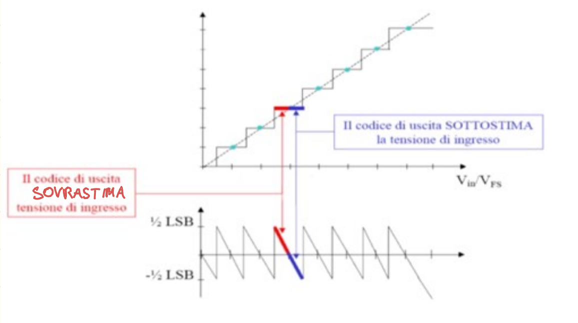
Numero di bit del Convertitore 4 bit → $2^4 = 16$ livelli

A parità di fondo scala, se codifico con un numero di bit più elevato (ovvero ho un maggior numero di livelli/quanti) ho la possibilità di creare una gradinata via via con "l'altezza" di ogni gradino minore; ottengo essenzialmente una retta inclinata (quasi lineare!).

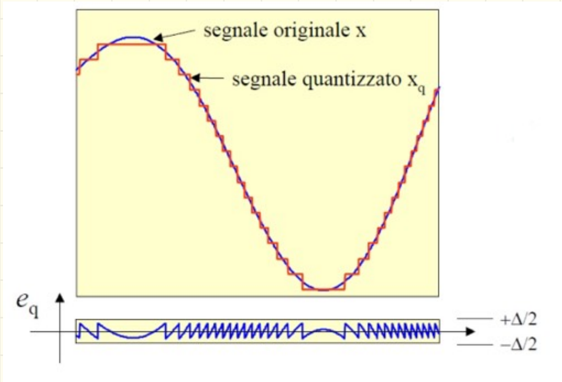
Se guardiamo la formula sopra, ci accorgiamo che (a parità di FS) all'aumentare del numero di bit del convertitore, la **risoluzione** diminuisce

ERRORE DI QUANTIZZAZIONE

$$\boxed{E = S_R - S_M}$$



Se consideriamo un segnale di tipo rampa (costante) e lo andiamo a quantizzare, **commettiamo un errore**. Questo perché il segnale si trova **per metà tempo al di sopra del livello e per metà tempo al di sotto del livello**. Quando invece il segnale si trova proprio al valore del livello **non commettiamo errore**. Di conseguenza il grafico dell'errore commesso è rappresentabile con una sorta di segnale a **dente di sega**: vale zero quando il segnale corrisponde al livello, e vale la metà dell'intervallo del livello quando il segnale è agli estremi.



Nel caso di una sinusoide l'errore non è più proprio un segnale a dente di sega ma gli si avvicina per la maggior parte del tempo. La differenza più importante la troviamo ai picchi della sinusoide: **in questi punti non riusciamo a trasmettere fedelmente il segnale in ingresso**: **<u>perdiamo una parte di informazione</u>**.

Ovviamente se aumentiamo il numero di bit utilizzati per quantizzare il segnale, diminuisce l'errore.

La risoluzione migliora al crescere di *b*.

Esempio con FS=10 V :

- **b=3 bit** ⇨ **LSB=1.25 V**
- **b=8 bit** ⇨ **LSB= 39 mV**
- **b=12 bit** ⇨ **LSB= 2.44mV**

ES: Con 16 bit e FS = 10 V abbiamo

$$Q_{16} = \frac{10V}{2^{16}} = \frac{10V}{65536} \cong 0.153\,mV = 153\,\mu V$$

NON SI SCRIVE MAI COSÌ

Si scrive così

Questo livello di precisione è elevatissimo: basta alitare sul circuito che stiamo misurando per far cambiare la temperatura quanto basta per far cambiare l'output. Di conseguenza è inutile andare oltre i 16 bit.

La fallacia è che al momento abbiamo computer a 64 bit, e **non riusciremo mai a sfruttarli Interamente per manipolare grandezze provenienti dal mondo reale** (analogico).

Questo è il **limite tecnologico** a cui siamo sottoposti. E' proprio per questo che siamo difronte ad un **bottleneck**.