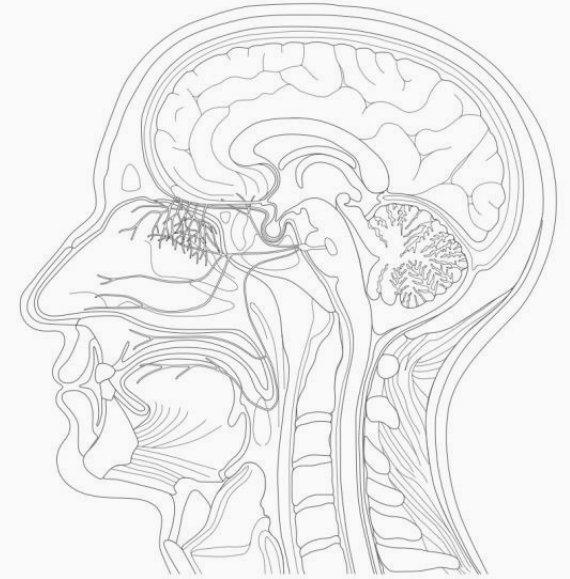


Conversione A/D



Tutor: Dr. Giovanna Nordio e Giulia Vallini

Prof. Mattia Veronese

Email: mattia.veronese@unipd.it

Dipartimento di Ingegneria dell'Informazione

Ricevimento: su appuntamento (e-mail)
Edificio DEI/A, piano 2o, stanza 214

Obiettivi del laboratorio

- Quantizzare un segnale dato un certo numero possibile di livelli e una strategia di approssimazione (arrotondamento o troncamento)
- Studiare come diverse tipi di interpolazioni permettono di approssimare l'andamento dei segnali nel tempo

Esercizio 1 – CHALLENGE

Si consideri un segnale sinusoidale con ampiezza di 1.5 V e un periodo di oscillazione di 100 secondi, campionato con una frequenza di 1Hz

- 1) Rappresentare graficamente il segnale sinusoidale nell'intervallo temporale da 0 a 150 secondi.
- 2) Definire due quantizzatori associati ad una codifica con segno a 4 bit, uno per arrotondamento (suggerimento: usare il comando **round**) e uno per troncamento (suggerimento: usare il comando **fix**), e calcolare il corrispondente errore di quantizzazione.
- 3) Visualizzare il segnale sinusoidale originale, e i due segnali quantizzati in un'unica figura.

Esercizio 1 – ISTRUZIONI DI CONSEGNA [Per prepararsi all'esame]

Consegnare un file .mat sul sito moodle del corso (sezione LABORATORIO) contenente:

- Una struttura '*quant*' che contenga:
 - quantizzatore ad arrotondamento (quant(1).data)
 - errore del quantizzatore ad arrotondamento (quant(1).err)
 - quantizzatore a troncamento (quant(2).data)
 - errore del quantizzatore a troncamento (quant(2).err)

Attenzione:

- Matlab è case sensitive (sbagliare il nome delle variabili equivale a sbagliare l'esercizio)

Esercizio 2

Data la funzione

$$y = 1/(1+x^2)$$

nota nei punti $x = 5 \cos(z)$, con z linearmente distribuito tra 0 e π in 11 campioni

- 1) Disegnare il valore della funzione in questi punti
- 2) Trovare una approssimazione della funzione in 100 punti equispaziati (suggerimento: usare il comando *linspace*) dell'intervallo $[-5, 5]$, usando l'interpolazione lineare e spline (usare il comando *interp1*).
Plottare sulla stessa figura (usare il comando *hold on*) del punto 1) le due interpolazioni.

Interpolazione in Matlab

Interpolazione in Matlab è spesso implementata tramite il comando **interp1**

$y_i = \text{interp1}(x, y, x_i, \text{metodo})$

che restituisce un vettore **yi** i cui elementi sono i valori delle ordinate interpolate ottenute in corrispondenza dell'ascissa **xi**, note le ascisse **x** e le ordinate **y** che concorrono a definire la curva interpolante.

Metodo di interpolazione:

- *linear*: interpolazione tramite spline cubiche
- *spline*: interpolazione tramite spline cubiche
- *cubic*: interpolazione con cubiche di Hermite
- *pchip*: interpolazione con cubiche di Hermite
- *nearest*: interpolazione al punto più vicino

Attenzione. Il vettore **xi** deve essere contenuto nel vettore **x**, altrimenti non si parla più di interpolazione ma di estrapolazione del segnale.