

Esercitazione 12

Wavelets

31 Maggio 2022

1. DFT e DWT

- (a) Considerare un segnale di dimensione 512 con tutti zeri ed un picco al campione 200 che vale 50. Effettuare la FFT del segnale e visualizzare il modulo dei coefficienti ottenuti. Ripetere lo stesso esercizio con un segnale con un picco a 100.
- (b) Il comando `cwt(x, scipy.signal.ricker, np.arange(1,48))` della libreria `scipy.signal` calcola la trasformata wavelet continua del segnale x , 'ricker' è il tipo di wavelet utilizzato (mexican hat) e `np.arange(1,48)` indica i livelli che si vogliono analizzare. Utilizzare il comando `cwt` per calcolare la trasformata wavelet continua dei due segnali introdotti precedentemente e fare un plot dei coefficienti ottenuti.
- (c) Costruire un segnale con due picchi molto vicini e calcolare la trasformata wavelet continua, visualizzare i coefficienti ottenuti e osservare come cambia il plot all'avvicinarsi dei due picchi.

2. DWT e compressione

- (a) Leggere il file `chirp.mat` (scaricabile dalla cartella del corso nel portale della didattica) utilizzando il comando `sio.loadmat('chirp.mat')['x']`. Fare un plot del segnale contenuto nel file.
- (b) Con il comando `dwt(x, 'db4')` della libreria `pywt` ottenere la scomposizione al primo livello del segnale. Il comando `dwt` restituisce in output due vettori contenenti i coefficienti dell'approssimazione e del dettaglio. Utilizzando il comando `upcoef` della libreria `pywt` ottenere i dettagli d e le approssimazioni a al primo livello.
- (c) Utilizzando il comando `idwt` della libreria `pywt` ricostruire il segnale. Confrontare il risultato ottenuto con la somma $d + a$, dove d e a sono i dettagli e le approssimazioni calcolate al punto precedente.
- (d) Utilizzare ora il comando `wavedec` della libreria `pywt` per avere livelli di decomposizione più precisi. Provare ad utilizzare tre livelli.

- (e) Provare a comprimere il segnale eliminando i coefficienti di dettaglio meno espressi, calcolare la media del valore assoluto dei coefficienti di dettaglio al terzo livello ed azzerare i coefficienti inferiori a tale valore. Ricostruire il segnale utilizzando il comando `waverec` della libreria `pywt`. Calcolare l'errore quadratico medio del segnale ricostruito.