Teoria ed elaborazione dei segnali

Esercitazione di laboratorio 1

Esercizio 1 (convoluzione lineare)

Dati i seguenti segnali tempo-discreto

$$x(n) = \begin{cases} \sin\left(\frac{\pi n}{5}\right) & 0 \leq n \leq 4 \\ 0 & \text{altrove} \end{cases}$$

$$y(n) = \begin{cases} 1 & 0 \leq n \leq 2 \\ 0 & \text{altrove} \end{cases}$$

calcolare la loro convoluzione lineare $z(n) = x(n) * y(n)$ *senza* usare la funzione di libreria `conv()`. Successivamente, verificare il risultato comparandolo con l'uscita della funzione `conv()`.

Suggerimento: In MATLAB, a differenza della maggior parte dei linguaggi di programmazione, gli indici degli array (*vettori* in gergo MATLAB) partono da 1 e non da 0.

Esercizio 2 (convoluzione circolare)

Calcolare la convoluzione circolare dei segnali precedenti $z_C(n) = x(n) \otimes y(n)$ *senza* usare la funzione di libreria `cconv()`. Successivamente, verificare il risultato comparandolo con l'uscita della funzione `cconv()`. Si ricorda che il periodo N della convoluzione circolare è un paramentro importante che influenza il risultato finale. Provare a calcolarla con valori diversi di periodo, da 5 a 10 campioni, e compararla con la convoluzione lineare $z(n)$ calcolata precedentemente. Dopo quale valore di N diventano identiche?

Suggerimento: Per "periodicizzare" un segnale discreto, può essere utile la funzione di libreria MATLAB `mod()`. Tuttavia, è importante tenere presente che `mod( ,N)` restituisce un numero compreso tra 0 e $N-1$, mentre MATLAB si aspetta come indice di un array un numero compreso tra 1 ed N. A tale scopo, un utile trucco per risolvere questo problema è aggiungere 1 fuori dal modulo, dopo aver tolto 1 dentro il modulo. Ossia: `mod( -1,N)+1`.

Bonus: La convoluzione circolare può essere implementata nel dominio della frequenza con la trasformata discreta di Fourier (DFT). Ri-scrivere la funzione scritta nel punto precedente utilizzando le funzioni di libreria MATLAB `fft()` e `ifft()`.

Bonus 2: La convoluzione circolare può essere implementata anche come prodotto di un vettore con una matrice *circolante*. Ri-scrivere la funzione precendente implementando la convoluzione circolare in questo modo.

Esercizio 3 (mutua correlazione e stima del ritardo)

Un segnale $x(n)$ di durata N campioni viene irradiato periodicamente dall'antenna di un trasmettitore. Un ricevitore mobile (che conosce il segnale $x(n)$) riceve una versione rumorosa e ritardata del segnale trasmesso

$$r(n) = x(n-D) + g(n) \tag{1}$$

dove D è un valore di ritardo (intero) e $g(n)$ è un segnale di rumore additivo gaussiano bianco con varianza σ^2. Il ricevitore è in grado di stimare la distanza dal trasmettirore tramite una stima del ritardo di propagazione D tramite la cross-correlazione tra $r(n)$ ed $x(n)$.

Implementare una funzione `r=mychannel(x,D,sigma)` corrisponente all'equazione (1). Partendo dallo script `Esercitazione13.m` e dalla funzione `mychannel()` produrre un grafico della stima del ritardo D per diversi valori di N e σ^2 sia nel caso in cui `configBit` (presente in `Esercitazione13.m`) corrisponde a 0 e 1.

Suggerimento: MATLAB ha una funzione di libreria `randn()` che genera un vettore di numeri casuali indipendenti distribuiti con una densità di probabilità Gaussiana a media nulla e varianza unitaria.