Elaborazione dei Segnali

Esercitazione "software" #2

Filtraggio di segnali musicali campionati

Scadenza per consegna: 13 gennaio 2025

Scopo dell'esercitazione

- Caricare due file musicali campionati (a vostra scelta) che abbiano una durata superiore ai 20 secondi
  - Vi consigliamo ovviamente di usare i due file usati per l'Esercitazione 1
    - E le stesse linee di codice per caricare il segnale in memoria

- L'obiettivo dell'esercitazione è di applicare un filtraggio lineare ai segnali di ingresso e di osservare i segnali di uscita
  1. Ascoltandoli ☺
  2. Plottando gli spettri risultanti (usando lo stesso codice già preparato per la esercitazione 1)

Brevi richiami teorici

Filtraggio numerico su segnali campionati

Filtraggio tramite convoluzione discreta

- Definizione della convoluzione discreta

$$y(n) = h(n) * x(n) = \sum_{k=-\infty}^{+\infty} x(k)\,h(n-k)$$

- In Matlab, la convoluzione discreta tra due vettori (di lunghezza finita) è implementabile direttamente tramite la funzione *conv()*
  - In Python, si possono facilmente trovare delle analoghe implementazioni

- SE *x(n)* e *h(n)* sono delle sequenze campionate di segnali a tempo continuo *x(t)* e *h(t)*, l'uscita *y(n)* è in generale una valida "approssimazione" della versione campionata di *y(t)*

Progetto di filtri numerici

- In realtà, la progettazione di un filtro numerico *h(n)* che approssimi al meglio il filtro *H(f)* nel dominio continuo è un argomento teorico abbastanza complesso
  - Che non abbiamo modo di trattare in questo corso ☹
- Tuttavia, è abbastanza intuitivo il seguente risultato: a patto che la frequenza di campionamento sia sufficientemente alta, una ragionevole approssimazione si ottiene come segue:
  1. Dato il filtro *H(f)* nel tempo continuo, si calcola la sua risposta all'impulso *h(t)*
  2. Si campiona *h(t)* alla stessa frequenza di campionamento del segnale di ingresso, e si usa il risultate *h(n)* come risposta all'impulso nel dominio discreto

Testo dell'esercitazione Software #2

Filtraggio numerico su segnali musicali campionati

Esercitazione software #2

- Caricare in memoria un file audio campionato *x(n)*
- Calcolare l'uscita di un filtraggio con le *h(n)* specificate alla slide successiva tramite convoluzione discreta

- Per ciascuna *h(n)*
  1. Ascoltare il segnale di uscita ☺
  2. Plottare lo spettro di uscita (ottenuto con le stesse tecniche della esercitazione 1) e confrontarlo qualitativamente con quello di ingresso
  3. Plottare la funzione di trasferimento risultante calcolandola come il rapporto tra lo spettro di uscita e quello di ingresso, calcolato per ciascuna delle frequenze discrete (si veda commento ultima slide)

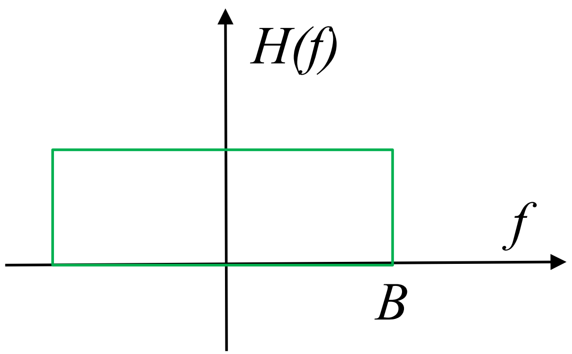
- Ripetere il tutto con il secondo file e confrontare i risultati

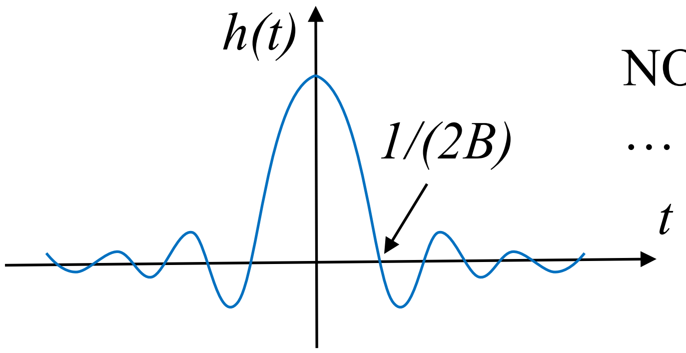
Risposte all'impulso *h(n)* da utilizzare

1. Versione discreta di una porta nel tempo continuo di durata *T* secondi
2. Versione discreta di una porta nella frequenza continua con una banda pari a *B* [Hz]
   - Si passi dal campionamento della *h(t)* che sarà proporzionale ad una *sinc(·)* con opportuni parametri
   - ATTENZIONE: dovete rendere causale il filtraggio con un opportuno ritardo (si veda slide successiva)
3. Versione discreta di un filtro passa-alto, con risposta in frequenza pari a 1-H(f), dove H(f) è una porta nella frequenza continua con una banda pari a *B* [Hz]

- *Suggerimenti: si noti che le prime due risposte sono di tipo passa-basso. Settare i parametri in modo che il taglio a 3dB sia approssimativamente attorno a 1 kHz*

Filtri causali

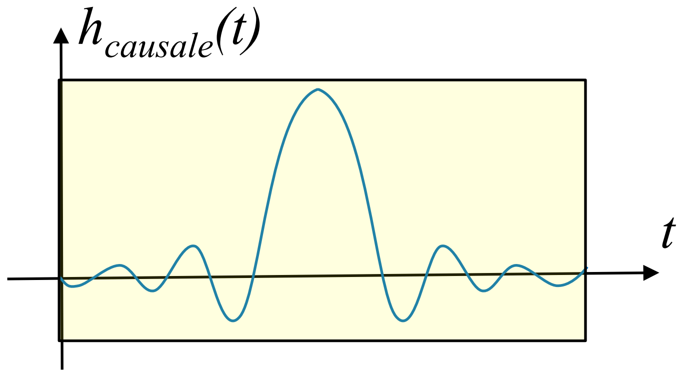




NON è causale!
… e non finisce mai ☺

Soluzione:

- ritardare
- poi troncare, prendendo un "numero sufficiente" di lobi laterali
- Poi campionare su questa risposta



Stima della risposta in frequenza

- Per l'ultima domanda: *"Plottare la funzione di trasferimento risultante calcolandola come il rapporto tra lo spettro di uscita e quello di ingresso, calcolato per ciascuna delle frequenze discrete"*

- Si chiede in sostanza di implementare numericamente la formula
  - Nel dominio delle frequenze discrete

$$H(f) = \frac{Y(f)}{X(f)}$$

- NOTA: purtroppo per frequenze dove $X(f) \approx 0$ numericamente il problema risulta mal posto, poiché si ha una situazione del tipo 0/0 che può dare risultati numericamente "casuali"

Stima della risposta in frequenza

- Per l'ultima domanda: *"Plottare la funzione di trasferimento risultante calcolandola come il rapporto tra lo spettro di uscita e quello di ingresso, calcolato per ciascuna delle frequenze discrete"*
- Per risolvere (parzialmente!) il problema della slide precedente:
- SOLO per la stima della risposta in frequenza, usare come segnale di ingresso un "rumore gaussiano bianco"
  - Verrà spiegato nell'ultima parte del corso a livello teorico
  - … ma per ora vi basta sapere che:
    1. Il rumore gaussiano bianco ha teoricamente uno spettro piatto in frequenza
    2. Si può generare in Matlab con la funzione *randn(·)* (o l'equivalente in Python)