Esercitazioni informatiche per il corso di Teoria dei Segnali - A.A. 2016-2017

Al fine di migliorare l'apprendimento degli argomenti affrontati `nel testo` dal punto di vista teorico, sono proposti i seguenti spunti di riflessione, da investigare mediante la scrittura di codice informatico, realizzato in forma di script da eseguire mediante gli ottimi `Gnuplot` e `Octave`. Di seguito sono indicate le tracce da cui partire, mentre nella sezione seguente sono riportati i riferimenti agli script realizzati nel corso dell'anno per affrontarte i temi proposti. Un modo pratico per ottenere TUTTO il codice sviluppato, è quello di scaricarlo in forma di pacchetto zippato, presso `http://infocom.uniroma1.it/alef/libro/stud/script.zip`.

1 - Risoluzione spettrale al variare della dimensione della finestra di analisi

A partire dal codice `cap3/f3.75.gnu`

1. impostare le due frequenze a 10 ed 11 Hz ($\Delta f = 1$ Hz);
2. rimuovere il codice relativo al plot con finestre `tau_2` e `tau_3`;
3. eseguire ripetutamente il grafico variando la dimensione di finestra `tau`, a partire da due secondi, e ridurla man mano, finché le due frequenze non sono più distinguibili; annotare il valore `tau` di quando accade, sia per la finestra rettangolare, che per quella triangolare;
4. ripetere l'esperimento impostando le due frequenze a 10 ed 20 Hz ($\Delta f = 10$ Hz), ed annotare il nuovo valore di `tau`;
5. verificare per quale valore del prodotto $\Delta f \cdot$`tau` la risoluzione spettrale si annulla nei due casi;
6. senza fare nuovi grafici, qual'è il minimo intervallo di frequenza Δf che è possibile risolvere nei due casi, avendo a disposizione un intervallo temporale di segnale `tau = 0.5` secondi?

2 - Calcolo della DFT di una sinusoide

A partire dal codice `cap4/f3.30a.m`

1. impostare una nuova frequenza per la sinusoide, ad esempio `freq = 7` Hz, lasciando gli altri parametri immutati;
2. eseguire il grafico della DFT a 64 punti, e commentare il risultato;
3. modificare il codice in modo che le condizioni di calcolo siano riportate in figura, ovvero `"Sinusoide a` *xx* `Hz campionata a` *yy* `Hz per` *zz* `sec pari a` *nn* `periodi"`:
4. calcolare la nuova frequenza di campionamento in modo da far entrare 2, 10, e 40 periodi nei 64 punti della DFT, e:
   (a) salvare su file i grafici corrispondenti;
   (b) commentare il risultato.

3 - Lettura, grafica ed ascolto di un file audio

Realizzare uno script Octave che

1. legga un file audio in formato `.wav`;
2. lo grafichi e faccia ascoltare per intero;
3. grafichi il segnale limitatamente a finestre di durata limitata, sovrapposte l'un l'altra del 50%;
4. effettui l'ascolto del contenuto delle singole finestre, concatenando più volte la relativa forma d'onda.

4 - Equivalenza tra media temporale e media di insieme

A partire da un file audio

1. stimare dinamica, valor medio, varianza e energia dei campioni, valutando le rispettive medie temporali;
2. calcolare l'istogramma ottenuto contando i campioni che rientrano tra le soglie ottenute suddividendo uniformemente la dinamica;
3. stimare la d.d.p. del segnale inteso come membro di un processo, a partire dall'istogramma;
4. valutare il momento di primo e secondo ordine come media di insieme, a partire dai valori dell'istogramma, e confrontare il risultato con le rispettive medie temporali.

5 - Filtraggio di un segnale

- eseguire la lettura di un file audio, annotando la relativa f_c;
- calcolare la risposta impulsiva $h(t)$ di un passabasso che filtra a partire da 1/8 della f_c (un quarto della massima);
- campionare $h(t)$ in h_n, e considerarne una versione *troncata* a + e - 80 campioni;
- aggiungere un ritardo in modo da avere un sistema causale;
- realizzare il grafico della h_n e quello della risposta in frequenza $H_m = DFT\{h_n\}$
- eseguire il filtraggio del segnale ed il suo riascolto.

6 - Filtraggio di un segnale - bis

A partire dal codice di `filtra_audio_conv.m`, generalizzare al caso di una frequenza di taglio qualsiasi, purché ovviamente $< f_c/2$. Sostituire quindi al loop della convoluzione, una chiamata a `fftfilt()`.

7 - Grafico di gaussiana bidimensionale

8 - Generazione di processo gaussiano, calcolo d.d.p., correlazione e densità spettrale, filtraggio

Facendo uso delle funzioni `randn` e `xcorr` (del package `signal`) di *Octave*, generare `N` campioni di rumore gaussiano bianco, e verificarne la d.d.p. gaussiana, l'autocorrelazione impulsiva, e la densità spettrale bianca. Ripetere le tre misurazioni dopo aver filtrato il processo mediante un filtro definito mediante i campioni della sua risposta impulsiva `h(n) = -0.5,0,1,0,-0.5`. Graficare infine la $|H(f)|^2$ del filtro, e le grandezze già calcolate, ma ora relative al segnale filtrato.

9 - Densità di probabilità all'uscita di un filtro

Al § 6.4.2 si mostra come filtrando un processo gaussiano, questa rimane gaussiano. Quindi, si *congettura* che se la banda del filtro è sensibilmente *minore* di quella del segnale, l'effetto di media dei valori di ingresso operati mediante una risposta impulsiva abbastanza lunga può produrre una *gaussianizzazione* del segnale in transito. Realizzare un script Octave per valutare cosa accade realmente.

10 - Effetto della distorsione di fase

Al § 7.2.2 si discute di come una risposta in frequenza per la quale la fase non rispecchi un andamento *lineare*, produca una distorsione che consiste in un diverso ritardo con cui le componenti frequenziali del segnale si presentano all'uscita del filtro, provocando una conseguente modifica della forma d'onda. Allo scopo di sperimentare questo fenomeno, realizzare uno script che

- sintetizzi una risposta in frequenza con modulo unitario e fase

    1. nulla
    2. lineare
    3. casuale

e grafichi modulo e fase di tale $H(f)$, assieme alla parte reale ed immaginaria della corrispondente risposta impulsiva $h(t)$ (tutte intese come forme campionate degli equivalenti continui). Quindi,

- generi un segnale da porre in ingresso nella forma di

  1. una coppia di sinusoidi
  2. un'onda quadra
  3. un segnale audio letto da disco

e produca, in una seconda finestra, sia il grafico del segnale originario, sia di quello filtrato mediante $h(t)$. Nel caso di segnale audio, verificare anche che, nonostante la forma d'onda sia cambiata, questo non causi alterazioni percettive.

11 - Modulazione FM

Al § 10.3 vengono illustrate le tecniche di modulazione angolare, e viene dichiarato che sebbene la densità spettrale dell'inviluppo complesso $\mathcal{P}_{\underline{x}}(f)$ debba essere valutata caso per caso per i diversi segnali modulanti $m(t)$, una regola comunemente adottata è quella della regola di Carson, ovvero $B_C \simeq 2W(\beta + 1)$, ottenuta per il caso di modulante sinusoidale a frequenza W, in cui $\beta = k_\phi$ o $\beta = {}^{k_f}/_{W}$ a seconda se PM o FM. Verifichiamo dunque quale dovrebbe essere il valore di β per mantenere valida l'approssimazione offerta dalla regola di Carson, nel caso in cui $m(t)$ sia un diverso segnale. Sviluppiamo pertanto uno script Octave che

- generi un segnale modulante $m(t)$ con frequenza massima W, di tipo

  – sinusoide ad ampiezza unitaria;
  – onda quadra ad ampiezza unitaria;
  – rumore gaussiano bianco;
  – rumore con d.d.p. uniforme tra + e - uno;
  – rumore con diversa d.d.p. $p_M(m)$ di tipo asimmetrico.

di cui visualizzare forma d'onda, spettro di potenza, istogramma delle ampiezze, ed andamento della fase modulante $\alpha(t)$ realizzata a partire da $m(t)$ nei casi di modulazione di fase (PM) o di frequenza (FM).

Quindi in una seconda finestra visualizzare, per diversi valori dell'indice di modulazione β,

- le componenti in fase e quadratura dell'invilupppo complesso del segnale modulato;
- il suo spettro di densità di potenza $\mathcal{P}_{\underline{x}}(f)$

Servirsi del simulatore sviluppato, per valutare la veridicità delle affermazioni

- nel caso in cui il segnale modulante sia un processo, l'indice di modulazione si ri-definisce come $\beta' = \frac{\sigma_f}{W}$;
- per β molto grande la densità di potenza $\mathcal{P}_{\underline{x}}(f)$ tende ad assumere la forma della d.d.p. $p_M(m)$

Risposte

Esercizio 1

Nei due casi si ottiene $\alpha = \Delta f \cdot \tau = 1.33$ e 1.66 per le finestre rettangolare e triangolare, dunque con `tau = 0.5` si risolvono $\Delta f = \alpha/$`tau = 2.66` e `3.32` Hz rispettivamente.

Esercizio 2

Si può notare che non compaiono più due singole righe spettrali, a causa del numero di periodi non più intero entro la finestra di analisi. Il codice si modifica aggiungendo le variabili

```
N = 64; # num.  punti DFT;
dur = N/rate; # durata in secondi;
nper = dur*freq; # num.  di periodi;
```

e creando una stringa da passare come secondo parametro alla funzione di `disegna` come

```
titolo = [ "Sinusoide a ", num2str(freq), " Hz campionata a ", num2str(rate),
" Hz per ", num2str(dur), " sec pari a ", num2str(nper), " periodi"];
```

La nuova f_c si ottiene considerando che il numero di punti/periodo risulta `N/nper`, e che in un secondo ci sono `freq` periodi, dunque è sufficiente calcolare un nuovo

```
rate = freq*N/nper
```

La figura si salva in formato pdf a partire dalla voce di menù "file", anche se a me non riesce sempre. Il commento è che quando `rate` scende al disotto della freq. di Nyquist di 14 Hz, si assiste al fenomeno *dell'aliasing*.

Esercizio 3

Il risultato è lo script di nome `audio.m`

Esercizio 4

Il risultato è lo script di nome `stima_stat.m`

Esercizio 5

Il risultato è lo script di nome `filtra_audio.m`, che nei commenti illustra i passi seguiti. *Siamo sicuri che sia giusto?* No, non lo è. Ci sono due problemi. Il primo, è che i valori del risultato sono *moooolto più grandi* di quelli di partenza. Il secondo, è che il grafico della FFT di `h(n)` appare *troppo concentrato* alle basse frequenze rispetto a quanto ci si aspettava, ovvero un passabasso a frequenza $f_c/8$: infatti la max freq. di una FFT è pari a f_c, dunque si sarebbe dovuto osservare un taglio a $1/8$ delle ascisse. Affrontiamoli uno alla volta.

Fattore di scala Allo scopo di approfondire l'analisi, si è sviluppato lo script `filtra_audio_conv.m` che anziché affidarsi a `fftfilt`, esegue la *convoluzione discreta* $y_n = \sum_{k=-\infty}^{\infty} x_k h_{n-k}$ (vedi § 4.5.1), che funziona correttamente, tranne appunto per *la dinamica dei valori di uscita*, che sono di gran lunga maggiori di quelli di ingresso... di quanto? Di un fattore pari alla frequenza di campionamento! Infatti, sviluppando per intero il passaggio solo accennato alla nota 43, i campioni dell'uscita sono pari a

$$
\begin{aligned}
y(nT_c) &= \int_{-\infty}^{\infty} x(\tau)\, h(nT_c - \tau)\, d\tau = \\
&= \int_{-\infty}^{\infty} \left[\sum_{k=-\infty}^{\infty} x(kT_c) \cdot \mathrm{sinc}\left(f_c(\tau - kT_c)\right) \right] \left[\sum_{j=-\infty}^{\infty} h(jT_c) \cdot \mathrm{sinc}\left(f_c(nT_c - \tau - jT_c)\right) \right] d\tau = \\
&= \sum_{k=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} x(kT_c)\, h(jT_c) \int_{-\infty}^{\infty} \mathrm{sinc}\left(f_c(\tau - kT_c)\right) \mathrm{sinc}\left(f_c(\tau - (n-j)\,T_c)\right) d\tau = \\
&= \frac{1}{f_c} \sum_{k=-\infty}^{\infty} x(kT_c)\, h((n-k)\,T_c) = \frac{1}{f_c} \sum_{k=-\infty}^{\infty} x_k h_{n-k}
\end{aligned}
$$

in cui alla seconda uguaglianza si è applicata la formula di ricostruzione *cardinale* $x(t) = \sum_{n=-\infty}^{\infty} x(nT_c) \cdot \mathrm{sinc}\left(f_c(t - nT_c)\right)$ e dunque $h(t-\tau) = \sum_{j=-\infty}^{\infty} h(jT_c) \cdot \mathrm{sinc}\left(f_c(t - \tau - jT_c)\right)$, entrambe valutate per $t = nT_c$; alla terza si è considerato che $\mathrm{sinc}(x)$ è una funzione *pari*, permettendo di scrivere $\mathrm{sinc}\left(f_c((n-j)T_c - \tau)\right) = \mathrm{sinc}\left(f_c(\tau - (n-j)T_c)\right)$, ed alla quarta si è applicata la proprietà di ortogonalità tra $\mathrm{sinc}(x)$ traslati di multipli di $1/f_c$ (vedi § 4.1.3), per cui l'integrale vale $T_c = 1/f_c$ solo quando $k = n - j$, ovvero $j = n - k$.

Pertanto, i campioni del risultato della convoluzione sono pari alla convoluzione discreta tra i campioni di ingresso e quelli della risposta impulsiva, *diviso la frequenza di campionamento!!*

La convoluzione discreta ora descritta viene eseguita dallo script `filtra_audio_conv.m`, che impiegando un tempo considerevole, opera su di una versione accorciata del file originario, ed alla fine visualizza prima il risultato con i valori *eccessivi*, e poi quello dopo la divisione per f_c. Infine, il risultato viene salvato nel file `audio/conv.vaw`, in modo da permettere di apprezzarne l'esito con il programma `Audacity`. In particolare, richiedendo la visualizzazione del *sonogramma* (spettro) si può verificare la riduzione di banda, mentre effettuando uno *zoom*, si può verificare il ritardo introdotto dal filtro.

Trasformata della h_n Tornando allo script *sospetto* `filtra_audio.m`, il problema del grafico della FFT di `h(n)` appare essere un banale errore di svista di programmazione, dato che osservando meglio, la FFT è calcolata sul vettore `x` contenente il segnale audio, e non su `h` con la risposta impulsiva! Dunque quello mostrato è lo spettro del segnale, ed in questo caso il risultato torna!

Esercizio 6

Il risultato dell'esercizio è riportato in `filtra_audio2.m`, in cui `fftfilt` viene invocata sia con due, che con tre parametri. Nel primo caso la convoluzione è realizzata mediante un prodotto in frequenza tra le `fft` di entrambi i vettori `x` ed `h`, mentre nel secondo caso il terzo parametro indica un numero di punti inferiore al totale, e la convoluzione è eseguita su segmenti con il metodo di overlap and add. Notare come il secondo metodo richieda un tempo tanto maggiore, quanto più sono brevi i segmenti utilizzati nell'overlap and add. Il tempo richiesto è valutato mediante le funzioni `tic()` e `toc()`.

Esercizio 7

La soluzione è realizzata nello script `gaussiana-bi.m.`

Esercizio 8

La soluzione è realizzata nello script `filtra-gauss.m`

Esercizio 9

La soluzione è realizzata nello script `ddp_uscita.m`. In realtà nell'istogramma che stima la d.d.p. si notano due componenti: oltre a quella del segnale, è presente un forte contributo legato ai periodi silenti, con segnale di ampiezza ridotta, che produce un picco elevato attorno ai valori vicini allo zero.

Esercizio 10

La soluzione è realizzata nello script `dist_fase.m`. E' da notare come anche ponendo la massima attenzione per realizzare $H(f)$ a simmetria coniugata, a causa degli arrotondamenti numerici la parte immaginaria di $h(t)$ non risulti esattamente nulla.

Esercizio 11

Lo script realizzato è `spettro_fm.m`, in cui i valori numerici delle costanti sono stati *tarati* e verificati utilizzando il segnale sinusoidale ad ampiezza unitaria, e la regola di Carson $B_C \simeq 2W(\beta+1)$ per il calcolo della banda. Il processo di modulazione avviene calcolando prima $\alpha(t) = \begin{cases} m(t) & PM \\ 2\pi\int_{-\infty}^{t} m(\tau)\,d\tau & FM \end{cases}$, e quindi eseguendo l'istruzione `invc = exp(i*beta*alfa)` in cui `alfa` contiene i campioni di $\alpha(t)$. Dato che nel caso sinusoidale β rappresenta la massima deviazione di fase, pari a $\beta = \begin{cases} k_\phi & \text{(PM)} \\ \frac{k_f}{W} & \text{(FM)} \end{cases}$, che nella generazione dei segnali abbiamo posto $k_\phi = k_f = 1$, e che il risultato dell'integrale di $m(t)$ cosinusoidale $\alpha(t) = 2\pi k_f \int_{-\infty}^{t} m(\tau)d\tau = \frac{k_f}{w}\sin(2\pi wt)$ determina per $\alpha(t)$ una *riduzione* delle ampiezze di un fattore W, per rendere conforme l'occupazione di banda nel caso FM con quella prevista dalla regola di Carson, si è resa evidente la necessità di *pre-moltiplicare* il valore di `beta` per `W`. In altre parole, se `beta=5`, l'argomento dell'`exp()` deve avere modulo 5 e non ${}^{k_f}/_{W} = {}^{1}/_{W}$, dunque `alfa` va moltiplicato per W. In questo modo la banda occupata dalla densità spettrale calcolata conferma molto bene il risultato previsto per B_c.

Si osserva quindi che l'occupazione di banda risultato della simulazione per un segnale modulante gaussiano con $\sigma_x = 1$ risulta notevolmente maggiore a quella prevista da B_C. Al contrario, i risultati teorico e sperimentale divengono comparabili qualora si ponga $\sigma_x = {}^{1}/_{2}$, e non $\sigma_x = {}^{1}/_{\sqrt{2}}$ come ci si sarebbe potuti aspettare, volendo equiparare la potenza del processo gaussiano con quello della sinusoide ad ampiezza unitaria, pari a ${}^{1}/_{2}$. Questo può trovare motivazione nel fatto che il 95.5% dei possibili valori di una v.a. gaussiana a media nulla è compresa entro una dinamica pari a $\pm 2\sigma_x$, ovvero se $\sigma_x = {}^{1}/_{2}$ entro la dinamica ± 1, come per il caso di modulante sinusoidale, e per la cui banda l'approssimazione B_C presenta più o meno lo stesso ordine di accuratezza (4%). Tali ragionamenti sono confermati dalla simulazione per $m(t)$ uniforme, in cui la dinamica e compresa tra ± 1 come per la sinusoide, e che pertanto presenta accordo tra B_C e simulazione senza necessità di modificare l'ampiezza di $m(t)$.

Per il caso di onda quadra di nuovo compresa tra ± 1, si osserva che la banda occupata dopo modulazione PM *non aumenta affatto!* Il motivo di ciò può essere apprezzato dopo aver osservato l'andamento di x_c e x_s: si presentano una costante, e l'altra oscillante come l'onda quadra, e ciò significa che $\underline{x}(t)$ si alterna tra due valori di fase. Ma dato che se $|\alpha(t)| > \pi$ la fase *wrappa*, all'aumentare dell'indice di modulazione corrisponde semplicemente una diversa coppia di fasi, sempre comprese tra zero e π. Pertanto, in questo caso $\underline{x}(t)$ può essere pensato come *somma* di due portanti *intermittenti*, con uguale frequenza e con una differenza di fase pari alla dinamica in cui oscilla $\alpha(t)\ mod\,\pi$, in cui l'intermittenza ha lo stesso periodo dell'onda quadra, e sincronizzate in modo da essere attive una alla volta. Per entrambe, la densità spettrale si ottiene per convoluzione in frequenza tra la densità della portante (un impulso) e quella dell'onda quadra (le armoniche dispari), e quest'ultima dipende solo dal periodo dell'onda quadra, e non da β.

Viceversa, la modulazione FM per $m(t)$ ad onda quadra determina un $\alpha(t)$ ad onda triangolare, ovvero l'alternanza di due toni la cui frequenza aumenta (o dimunuisce) con β.

Questo appare anche come una conferma che $\mathcal{P}_{\underline{x}}(f)$ tende ad assumere la forma della d.d.p. $p_M(m)$, congettura che appare confermata osservando la similitudine tra l'istogramma di $m(t)$ *gaussiano* e la relativa $\mathcal{P}_{\underline{x}}(f)$: l'istogramma non risulta mai *esattamente* gaussiano, ma le sue *particolarità* sono riprese dalla $\mathcal{P}_{\underline{x}}(f)$ simulata. La congettura non è invece confermata qualora si adotti una d.d.p. poissoniana, ma in tal caso, osservando la $\mathcal{P}_{\underline{m}}(f)$, si nota che decade l'ipotesi di processo bianco, ovvero di campioni incorrelati.