BIOMEDICAL SIGNAL PROCESSING
05-03-2012

ECG: segnale quasi periodico deterministico
↳ è uno dei pochi

[periodico = si ripete uguale a se stesso a istanti definiti]

segnale deterministico = la cui evoluzione è in qualche modo determinato, prevedibile

↳ il contrario sono i segnali casuali

l'ecg si ripete circa uguale a se stesso a intervalli regolari

solo l'ecg viene compresso (veniva, ora non più) → registrazione holter

→ algoritmi di compressione che registravano solo pezzi particolari

Oggi non serve più la compressione.

EEG: segnale casuale (è generato dai neuroni, che sono organizzati in reti molto complesse e si depolarizzano ad istanti casuali)

POTENZIALI EVOCATI: risposta di un gruppo specifico di neuroni (vista, udito, movimento ...)

ad uno stimolo uguale rispondono allo stesso modo → segnale deterministico

↳ un gruppo di neuroni risponde in modo specifico: se non lo fa significa che ci sono dei problemi (però il loro segnale è molto piccolo rispetto al resto dei segnali dei neuroni)

EMG: segnale prodotto dai muscoli, segnale casuale

ma con elettrostimolazione diventa un segnale deterministico

} segnale bivalente

↳ analisi spettrale legata alla fatica muscolare

06-03-2012

Con il segnale eeg si vedono macro cose (es. epilessia → attacco epilettico)

ha un contenuto spettrale che varia con lo stato del soggetto

ECG: aritmie
EEG: epilessia, disturbi del sonno
POTENZIALI EVOCATI: sclerosi multipla
EMG: tunnel carpale, distrofia muscolare

RUMORE

per prelevare un segnale avrò qualcosa di applicato → registro segnale + rumore

$$x(t) = \underbrace{s(t)}_{\text{segnale}} + \underbrace{n(t)}_{\text{rumore}}$$

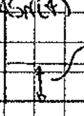
MODELLO ADDITIVO (il rumore si somma al segnale nel tempo)

Normalmente il rumore $n(t)$ è un processo casuale → rumore gaussiano bianco

: gaussiano = funzione di autocorrelazione impulsiva (delta di dirac nello zero)

↳ le ampiezze dei campioni di rumore (nel tempo) sono distribuite come una gaussiana

· bianco = lo spettro in frequenza è una linea piatta



$S_n(f)$ proporzionale alla potenza del rumore e quindi alla varianza σ_n^2

→ la funzione di autocorrelazione è nulla tranne che nell'origine: il rumore gaussiano bianco è scorrelato

autocorrelazione nulla ⇔ spettro piatto

Quantificare la qualità della registrazione: rapporto segnale-rumore.

↳ mette in relazione un parametro del segnale con un parametro del rumore

(il parametro dipende dal tipo di segnale / rumore)

segnale — deterministico — rumore — deterministico / casuale
segnale — casuale — rumore — deterministico / casuale

rumore deterministico: interferenza di rete.

Se ho due sorgenti di rumore calcolo il SNR nei due casi (devo dire a quale mi riferisco)
↳ devo specificare come abbiamo calcolato il SNR

Per fare un'operazione di filtraggio deve essere chiaro di quanto miglioro il SNR

- segnale casuale / rumore casuale — sempre con il modello additivo — es. slide 15 muscolo extracl

$$SNR = \frac{P_s}{P_m} = \frac{\sigma_s^2}{\sigma_m^2} \qquad SNR_{dB} = 10 \log_{10} \frac{P_s}{P_m} = 20 \log_{10} \frac{\sigma_s}{\sigma_m}$$

nell'esempio: segnale+rumore seguito da rumore

  - potenza del rumore: varianza campionaria dei campioni tra 1000 e 2000
  - potenza del segnale: varianza dei campioni 0-1000 meno varianza 1000-2000
    (ipotesi che il rumore ha la stessa potenza quando il muscolo è attivato che quando non lo è → segnale e rumore scorrelati
    senza questa ipotesi ad oggi non esistono soluzioni)

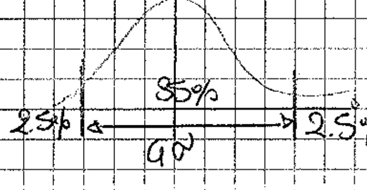
- segnale quasi periodico / rumore casuale — ecg / tremore muscolare

  la potenza del segnale dipende dalla finestra di osservazione (e quindi dalla frequenza)
  ↳ conviene calcolare un rapporto di ampiezze.

$$SNR = \frac{A_s}{4\sigma_m}$$

A_s ~ ampiezza di picco, valore efficace, ...
$4\sigma_m$ ~ 4 volte la deviazione standard

  - si sceglie A_s a seconda di cosa devo fare: es. ecg per la frequenza mi basta il valore picco-picco dell'onda R, oppure per altre cose il valore efficace, ...
  - con $4\sigma_m$ si cerca un intervallo di confidenza



$z_{95\%} = 1.96 \approx 2$ da entrambe le parti: 4
[il 4 può anche essere tralasciato, tanto si paragona il SNR prima e dopo il filtro, e lo si calcola nello stesso modo]

$$SNR_{dB} = 20 \log_{10} \frac{A_s}{4\sigma_m}$$

nell'esempio: ampiezza del rumore σ_m calcolata nel tratto isoelettrico tra T e P

- segnale quasi periodico / rumore periodico — ecg / rumore rete

$$SNR = \frac{A_{pp}}{N_{pp}}$$

bisogna usare lo stesso estimatore (piccopicco - picco - valore efficace)

$$SNR_{dB} = 20 \log_{10} \frac{A_{pp}}{N_{pp}}$$

- segnale casuale / rumore periodico

$$SNR = \frac{\sigma_s}{N_{rms}} \approx \frac{4\sigma_s}{N_{rms}} \qquad SNR_{dB} = 20 \log_{10} \frac{\sigma_s}{N_{rms}}$$

N_{rms} ↳ valore efficace

caso meno utilizzato: segnale a banda larga con rumore a banda stretta
↳ importa poco (magari è anche fuori banda) ↳ eeg: banda < 30Hz

es. $SNR_{dB} = 10\ dB$ (minimo, scarsa qualità)

$\Rightarrow \frac{P_s}{P_n} = 10$

si fa fatica a lavorare con SNR < 10-12 dB, accettabili 12-15 dB

> 15 dB buone registrazioni

FILTRAGGIO: eseguire operazioni su $x(t)$ per attenuare $n(t)$ senza toccare $s(t)$
↳ DENOISING

FILTRO: dispositivo in grado di attenuare opportune frequenze ↳ non posso cancellarle

ⓐ segnale e rumore occupano bande diverse e ben separate
↳ il muscolo scorre sulla cute dove ci sono gli elettrodi
artefatto da movimento e EMG ⇒ filtro PASSAALTO (10-20Hz)
↳ da pochi Hz a 15-20 Hz ↳ da 20 Hz in su

segnale ECG e tremore muscolare: c'è sovrapposizione, ma va bene
↳ 0.1Hz-125Hz ↳ 25-500Hz
un filtro PASSABASSO (125Hz)

segnale EEG e interferenza di rete: filtro PASSABASSO (35-60Hz)
↳ fino a 35Hz ↳ 50-60Hz

ⓑ segnale e rumore sovrapposti anche in frequenza, ma segnale a banda larga e rumore a banda stretta (o viceversa)

es. ECG e interferenza di rete ⇒ filtro RIGETTABANDA (in questo caso tocca anche il segnale, ma va bene)

Nel caso contrario ⇒ filtro PASSABANDA (intorno al segnale)

ⓒ segnale e rumore sovrapposti in frequenza a banda larga (paragonabile)
→ posso solo ricorrere a tecniche di filtraggio speciali che si adottano solo a segnali particolari (i segnali devono rispettare delle prop.

TECNICA DELL'AVERAGING

Si utilizza quando falliscono le tecniche tradizionali.

Segnali limitati nel tempo sommersi dal rumore → potenziali evocati (visivi, uditivi, somatosensoriali)

potenziale evocato: visivo = scacchiera che cambia colore, flash

uditivo = bip, click

somatosensoriale = stimoli caldo/freddo, riflessi sul piede

CERVELLO

funzioni nobili

due emisferi che controllano la parte opposta. Ogni emisfero è compartimentato

- zona frontale = associativa → determina il carattere (associare gli stimoli esterni alla tua reazione)

zona che prende la tua memoria e la proietta nella realtà; zona dove nascono le idee

- corteccia somatosensoriale (dietro la frontale) : emozioni, ...
- corteccia motoria (dietro) : es. servono + neuroni per il pollice che per il piede
- zona posteriore : ci sono i centri della vista → è dietro perché è molto importante e deve essere sicuro (osso + spesso) → 85% delle info
- zona temporale (dietro l'orecchio) : centro dell'udito

  ↳ prima e dopo ci sono centro del linguaggio (per capire) e delle parole (per parlare)

I gruppi di neuroni, se funzionano bene, rispondono → i potenziali evocati hanno un andamento noto e ripetibile (deterministico). Ma sono pochi i neuroni : il segnale del potenziale evocato è sepolto dal rumore (= processo casuale con caratteristiche simili all'eeg)

Per applicare la tecnica dell'averaging devo essere sicuro che il segnale sia presente → do tanti stimoli e registro tante risposte (ripeto per un certo numero di volte)

IPOTESI:

- segnale deterministico che si ripete nel tempo in modo prevedibile
- segnale deterministico si ripete un numero alto di volte in modo identico a se stesso (quasi : non sono tutti uguali → fatica : il cervello umano è fatto per rispondere a stimoli diversi, non uguali e ripetuti)

  ↳ non bisogna ripetere troppe volte ed incorrere in fenomeni di fatica

  ↳ il cervello si adatta : quando capisce, non attiva + neuroni → cambia la risposta

  ⇒ al massimo poche centinaia di stimoli alla distanza di 1 s

  → funzione di crosscorrelazione = 0 (nulla sempre)
- il rumore è scorrelato dal segnale, stazionario in senso lato (ha caratteristiche spettrali indipendenti dal tempo) ↳ è un processo casuale → tutti i momenti statistici indipendenti dal tempo → ma in natura non esistono segnali stazionari : in senso debole / lato (WSS) (momenti statistici solo di ordine 0 e 1 : media e varianza indipendenti dal tempo) ; ergodico (= quando le caratteristiche spettrali di un campione sono significative dell'intera popolazione)

  per un segnale reale la stazionarietà è molto difficile da dimostrare → bisogna vedere, e ergodicità → sempre è impossibile ⇒ si assume stazionario ed ergodico

Faccio la media di tutte le epoche che ho registrato.

- il segnale mediato è uguale al segnale
- il rumore mediato diminuisce

N: n° di epoche

→ il segnale è sempre lo stesso

$x_i(t) = s(t) + n_i(t)$ $\qquad$ $\bar{x}(t) = \frac{1}{N}\sum_{i=1}^{N} x_i(t) = s(t) + \frac{1}{N}\sum_{i=1}^{N} n_i(t)$

→ statisticamente la varianza del rumore mediato diminuisce di un fattore N

$$E[\bar{x}(t) - s(t)]^2 = \frac{1}{N^2} E\left[\sum_{i=1}^{N} n_i(t)\right]^2 = \frac{1}{N^2}\sum_{i=1}^{N} E[n_i(t)]^2 = \frac{1}{N^2} N \sigma_N^2 = \frac{\sigma_N^2}{N}$$

→ rapporto segnale rumore di una qualsiasi epoca

$SNR_0 = \frac{A_S}{4\sigma_N}$ $\qquad$ $SNR_A = \frac{A_S}{4\sigma_N/\sqrt{N}} = \frac{A_S}{4\sigma_N}\sqrt{N}$ $\qquad$ il rapporto segnale rumore è aumentato di $\sqrt{N}$

↳ dopo che ho applicato l'averaging

$100 < N < 200$

Vorrei lavorare con N più alto possibile, ma oltre ad un limite ho problemi di fatica, adattamento

La tecnica dell'averaging viene usata per determinare il numero di epoche per avere un certo SNR. => è una tecnica PILOTATA

Le tre ipotesi consentono di sapere di quanto aumenta il SNR.

- ad es. se il rumore non è stazionario, posso applicare la tecnica dell'averaging, il segnale (registrato) migliora, ma non so di quanto

• la tecnica non è applicabile solo se il segnale non è deterministico; le altre ipotesi sono di supporto per avere prestazioni note.

PROBLEMA DEL JITTER (latenza = tempo che intercorre tra qualcosa e qualcos'altro)
↳ tra stimolo e risposta nel p.e.

la latenza è una variabile casuale: le risposte non arrivano dopo lo stesso tempo. (sono uguali ma sfasate)

$$x_i(t) = s(t-t_i) + n_i(t)$$

In presenza di jitter, con la tecnica dell'averaging vado a mediare anche il segnale

FARE UNA MEDIA = FARE UN FILTRO PASSABASSO

-> è importante la varianza della latenza: più è piccola meglio è

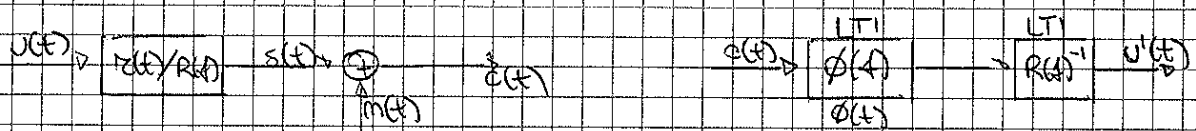
=> ci sono degli algoritmi di riallineamento

30: senza riallineamento non si vede il potenziale

FILTRO OTTIMO filtro WIENER

Si utilizza per trovare l'approssimazione migliore possibile del segnale dal segnale registrato senza apporre ipotesi.

Supponiamo di avere il potenziale biologico $u(t)$; elettrodi applicati al sistema hanno una loro funzione di trasferimento $R(f)$ (trasf. di Fourier) [$r(t)$ risposta all'impulso]



$$s(t) = u(t) \otimes r(t)$$
$$S(f) = U(f) \cdot R(f)$$
$$c(t) = s(t) + n(t)$$

L'obiettivo del filtro ottimo è ottenere la miglior stima possibile di $u(t)$ partendo da $c(t)$ => ottenere un filtro + deconvoluzione

$$U'(f) = C(f) \cdot \Phi(f) \cdot \frac{1}{R(f)}$$

Parlando di ottimizzazione devo fissare un criterio: minimizzare l'errore quadratico

$$e = \int_{-\infty}^{+\infty} [u'(t) - u(t)]^2 dt = \int_{-\infty}^{+\infty} [\hat{U}(f) - U(f)]^2 df = \int_{-\infty}^{+\infty} \left[\frac{[S(f)+N(f)]\Phi(f)}{R(f)} - \frac{S(f)}{R(f)} \right]^2 df$$

per minimizzare 'e' devo fare la derivata = 0 (è difficile fare la derivata di questo funzionale nel dominio del tempo). Scelgo di passare nel dominio della frequenza (TEO di PARSEVAL = le quantità energetiche si possono calcolare equivalentemente nel tempo e nella frequenza)

-> posso passare nel dominio della frequenza (il minimo non è lo stesso, ma la forma è lo stesso → a noi interessa quello, ...)

$e = \int [R(f)]^{-2} \{ |S(f)|^2 \cdot |1-\phi(f)|^2 + |N(f)|^2 |\phi(f)|^2 \} \, df$ deriva rispetto a $\phi(f)$

La soluzione è $\phi(f) = \dfrac{|S(f)|^2}{|S(f)|^2 + |N(f)|^2}$

Sono in grado di calcolare la funzione di trasf. di un filtro ottimo senza conoscere v e r

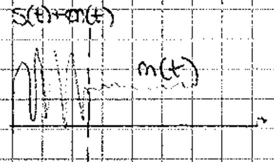
$c(t) = s(t) + n(t)$ $\quad$ $C(f) = S(f) + N(f)$ $\quad$ $|C(f)|^2 = |S(f)|^2 + |N(f)|^2 + 2|S(f)||N(f)| *$

coeff. di correlazione

Se 's' e 'n' sono scorrelati $|C(f)|^2 = |S(f)|^2 + |N(f)|^2$

Ma mi serve anche $|S(f)|^2$:

- se il segnale è presente solo in una parte:



$s(t) + n(t) - n(t)$

solo per segnale PULSATO

- METODO GRAFICO: disegno lo spettro di $C(f)$

ipotesi: ogni segnale biologico è limitato in banda: in alta frequenza c'è solo rumore

interpolo questa parte e la riporto su tutte le frequenze → ottengo $|N(f)|^2$

posso ottenere $|S(f)|^2 = |C(f)|^2 - |N(f)|^2$

TRASFORMATA Z

$x[n] \rightarrow \boxed{h[n]} \xrightarrow{y[n]}$

Ponendo all'ingresso $x[n] = z^n$ con z numero complesso

$$y[n] = \sum_{k=-\infty}^{+\infty} \overbrace{x[n-k]\,h[k]}^{\text{convoluzione}} = \sum_{k=-\infty}^{+\infty} z^{n-k} h[k] = z^n \underbrace{\sum_{k=-\infty}^{+\infty} z^{-k} h[k]}_{H(z)} = z^n H(z)$$

non è detto che sia sempre definita

$H(z)$ → trasformata z = funzione di trasferimento

LA TRASFORMATA Z DI UNA QUANTITÀ DISCRETA È UNA QUANTITÀ CONTINUA

RITARDATORE DISCRETO DI K PASSI

Sistema lineare tempo invariante che provoca un uscita $y[n] = x[n-k]$

$y[n] = z^{n-k} = z^n \cdot z^{-k} = z^n (H(z))$ $\quad$ $H(z) = z^{-k}$

è la base dei filtri digitali

$x[n] \rightarrow \boxed{h_1[n]} \xrightarrow{y_1[n]} \boxed{h_2[n]} \xrightarrow{y_2[n]}$

Calcolare $h_{eq}[n]$ ⇝ conviene usare $x[n] = \delta[n]$

$H_{eq}(z)$ ⇝ conviene usare $x[n] = z^n$

REGIONE DI CONVERGENZA = valori di z per cui la serie non diverge (la trasformata z non diverge)

è una regione circolare (una corona circolare)

↳ se la sequenza è causale, è la zona esterna al cerchio ($R^+ \to \infty$)

↳ se $|z|=1$, la regione è il cerchio di raggio unitario (es. $z=e^{j\omega}=e^{j2\pi k}$

$z = e^{j\omega} = \cos\omega + j\sin\omega \Rightarrow |z| = \cos\omega^2 + \sin^2\omega = 1$)

La trasformata di Fourier è un caso della trasformata z con $z=e^{j\omega}$

FILTRI DIGITALI

Sistema lineare tempo invariante con una f.d.t. ..... copiare da Lindo

$$H(z) = \frac{Y(z)}{X(z)} = \frac{b(1) + b(2)z^{-1} + \dots + b(nb+1)z^{-nb}}{1 + a(2)z^{-1} + \dots + a(na+1)z^{-na}}$$

- nb e na non è detto che sono uguali
- ci sono solo numeri e ritardatori discreti ⇒ polinomi
- a e b sono i coefficienti del filtro
- per specificare in modo univoco il filtro bastano a e b

$$b = [1 \ -1 \ 0 \ 3] \qquad H(z) = \frac{1 - z^{-1} + 3z^{-3}}{1 - 2z^{-3}}$$
$$a = [1 \ 0 \ 0 \ -2]$$

PROPRIETÀ

- la trasformata z è lineare
- la trasformata z è invertibile (bisogna dire qual è z)

$Y(z) = X(z) \cdot H(z)$

$H(z) = \frac{B(z)}{A(z)} \qquad A(z) \cdot Y(z) = B(z)\, X(z)$

è possibile passare al filtro digitale nel dominio del tempo

$$Y(z)(1 - 2z^{-3}) = X(z)(1 + z^{-1} + 3z^{-3})$$

↓ antitrasformo

$$y[n] - 2y[n-3] = x[n] + x[n-1] + 3x[n-3]$$
$$y[n] = x[n] - x[n-1] + 3x[n-3] + 2y[n-3]$$

il filtraggio è una combinazione di numeri ⇝ ci sono routine (di Matlab) che lo fanno automaticamente

- se nb = 0 il numeratore è una costante → il filtro è "ALL-POLE" di tipo RICORSIVO e AUTOREGRESSIVO (l'uscita ricorre su se stesso)
  ↳ filtro IIR (= risposta all'impulso infinita) ↝ transitorio di durata infinita (forma della risposta all'impulso)
- se na = 0 il denominatore è una costante → funzione "ALL ZERO", di tipo NON RICORSIVO, A MEDIA MOBILE (l'intervallo di campioni di cui si ha bisogno scorre)
  ↳ filtro FIR (= risposta all'impulso finita)
- na ≠ 0 e nb ≠ 0, funzione "POLE-ZERO", ricorsivo, autoregressivo e a media mobile
  ↳ filtro ARMA
  ↳ filtro IIR (predominano i poli sugli zeri)

Per dire l'ordine di un filtro ARMA devo dire due numeri

12 · 03 · 2012

Il primo elemento del vettore < a > vale sempre uno → tutti i coefficienti a e b sono rinormalizzati ad a_1

STRUTTURA A MINIMO NUMERO DI RITARDATORI

struttura che consente di schematizzare H(z) con il minimo numero di ritardatori

▽ è un moltiplicatore   sopra: MA   sotto: AR

Nello schema mettiamo il primo campione x[1] → va avanti fino al primo ritardatore, quando all'ingresso entra x[2]

$y[1] = b(1)\,x[1]$

$y[2] = b(1)\,x(2) + b(2)\,x(1) - a(2)\,y(1)$

TRANSITORIO DI CARICA DEL FILTRO: finché non ho inserito ma/mb campioni

↳ per tutti i filtri: i primi campioni filtrati non vanno considerati

Quando arriva il primo campione i valori di z valgono 0 (sono indeterminati)
↳ è una nostra imposizione
↳ sono i VALORI INIZIALI DEL FILTRO: possono essere caricati come vogliamo

FILTRAGGIO A BLOCCHI

segnale lungo 10.000 campioni → devo caricarlo: allocare memoria

lo filtro: devo allocare in memoria a e b → e poi devo allocare i campioni filtrati

Per un pc non c'è un problema, ma per dei microcontrollori il problema si pone

⇒ si filtra a blocchi: si carica e filtra un blocco di segnale alla volta

(tanto mi basta avere un n° di campioni pari all'ordine del filtro)

Per evitare che ogni blocco filtrato inizi con un transitorio si propagano i pesi dalla fine del filtraggio del blocco (m-1) all'inizio del filtraggio del blocco m → il filtro è precaricato (ZI)

matlab: [y, Zf] = filter(b, a, segnale, Zi)

(segnale filtrato; valori dei pesi finali; coefficienti del filtro (vettore); ↳ eventualmente i pesi)

es. filtrare ecg → è un segnale casuale → metto dei pesi iniziali casuali

⇒ ho sempre un transitorio, non serve

[l'ordine di un filtro = n° dei coefficienti − 1]

b è quasi sempre simmetrico o antisimmetrico

matlab ha anche delle funzioni per calcolare i coeff. a e b data delle specifiche

(matlab considera l'asse delle freq. tra 0 e 1 (tra 0 Hz e $f_c/2$ Hz))

butter (ordine, freq. taglio)   → normalizzato rispetto a $f_c/2$ 'passa basso'; se voglio passa alto → 'high'

freqz (b, a, ...)   rappresentare il diagramma di bode della funzione di trasferimento (modulo e fase)

↳ da z a f   ↳ servono per denormalizzare l'asse delle frequenze

Cerchiamo di avere una fase lineare → es. negli amplificatori, se sfaso le armoniche distorco

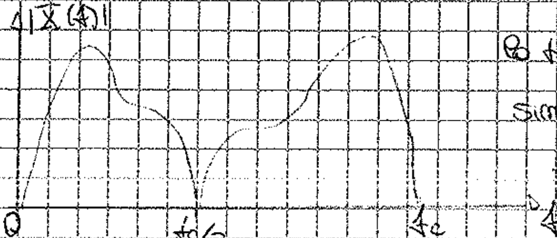
↳ è aspetto temporale del segnale

Anche per il filtraggio: un filtro a fase non lineare DISTORCE il segnale nel dominio del tempo

Ma i filtri a fase non lineare si usano:

- la distorsione in alcuni casi si può correggere
- per i processi casuali non ha importanza (es. emg e eeg)

→ ad es. invece per ecg e potenziali evocati è fondamentale non distorcere



la trasformata di Fourier è simmetrica (si può rappresentare simmetrica rispetto a $fc/2$ o a 0)

→ se la banda del segnale è $> fc/2$ ⇒ ALIASING

$fc/2 = f_N$ frequenza nyquist

matlab: freqz genera l'asse delle frequenze tra 0 (continua) e 1 ($= fc/2$)

freqz (b, a, n° punti, fcq. di campionamento) ← × disegnare il grafico

↳ rappresenta fino a 1000 ($fc/2$)

La trasformata di Fourier (anche di una quantità discreta) è continua, perciò posso rappresentarla su quanti punti voglio → devo inserire questi parametri

Controllare sempre il risultato → matlab può generare filtri sbagliati se ad es. specifiche alte e ordine basso

FILTRI IIR

Hanno risposta all'impulso infinita ⇒ sia filtri AR che ARMA

- consentono di soddisfare delle specifiche con un ordine minore dei filtri FIR
  ↳ transitorio minore, meno memoria, + veloce (− operazioni da fare)
- non hanno mai fase lineare, e la distorsione di fase non è nota
  ↳ si prova a recuperare

<u>FILTRO ANTICAUSALE / A DOPPIA PASSATA</u>

$H(z)$ funzione di trasf. non lineare

→ è come se inverto la fase, sommo 2 a φ

→ se $x[n] \to X(z)$ e inverto $x[n]$ ⇒ $x[n]_{inv} \to X(1/z)$ ← giro il tempo

giro il tempo ⇒ z^{-1}

$$X(z) \xrightarrow{H(z)} X(z)H(z) \xrightarrow{\text{time reverse}} X(1/z)H(1/z) \xrightarrow{H(z)} X(1/z)H(1/z)H(z) \xrightarrow{\text{time reverse}}$$

$$Y(z) = X(z)\,H(1/z)\,H(z) = X(z)\,|H(z)|^2$$

la fase del modulo quadro vale 0 → non distorce la fase

Posso annullare l'errore di fase, ma + costa di più

↘ non si può implementare in tempo reale

È possibile fare il filtraggio anticausale a blocchi ??

Condizioni: 15-03-2012

- segnale a fase non lineare (MAI se è lineare)
  ↳ filtro IIR
- segnale non deve essere distorto nel dominio del tempo

→ almeno due volte avanti ed indietro

Le condizioni sono sempre sul filtro, su $H(z)$

Non si può filtrare campione per campione (in tempo reale) ⇝ ma sui dispositivi

- La lunghezza della sequenza deve essere almeno tre volte l'ordine del filtro ricorsivo
  ↳ all'inizio del filtraggio si crea un transitorio: se la sequenza è troppo corta ho solo transitorio (in questo caso ce l'ho all'inizio e alla fine)
- La sequenza d'ingresso deve andare a zero all'inizio e alla fine (è teorico: va bene anche se non va a zero ma hanno lo stesso valore)
  [↳ condizione necessaria ma non sufficiente perché il filtro sia causale]
  → transitorio di carica = a quello di scarica ⇒ stesso transitorio all'andata e al ritorno

ESEMPIO

segnale : $s(t) = \sin(2\pi t \cdot 3)$
rumore : $n(t) = 0.25 \sin(2\pi t \cdot 60)$
⇒ fare un filtro passabasso

$f_c = 100\ Hz$ $\qquad T = 1s$ finestra temporale

MATLAB :

```
fc = 100
T = 1
t = 0 : 1/fc : T-1/fc      oppure   t = 1/fc : 1/fc : T
s = sin (2*pi*3*t)
n = sin (2*pi*60*t) * 0.25
x = s + n
```

→ sempre filtro ARMA con num/den dello stesso ordine

Adesso facciamo il filtro (usiamo butter ma va bene qualsiasi filtro)

```
[b, a] = butter (9, 10/(fc/2))      (50)
```

controllo il filtro : `freqz (b, a, 256, fc)` → l'attenuazione è tanta
→ la fase non è lineare

proviamo a filtrare normalmente : `y = filter (b, a, x)`

stampando y, vediamo che il disturbo è stato filtrato, non è + sincrono all'ingresso (c'è il transitorio), ma la sinusoide è distorta, non è una sinusoide

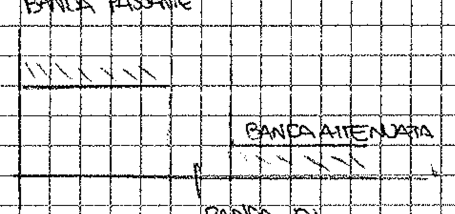
filtriamo a doppio passato : `yy = filtfilt (b, a, x)`

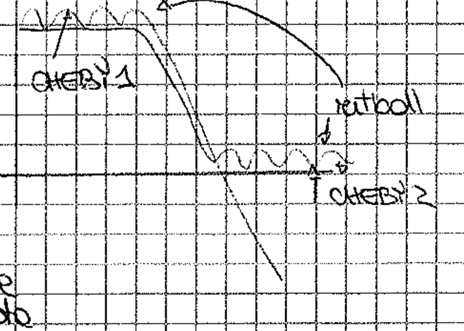
recupero anche il transitorio, e non è distorto

↳ es. richiede anche i db del ripple

MATLAB : filtri IIR → butter, cheby1, cheby2, ellip ⇒ prendono la funzione analogica e la traducono in digitale

FILTRO





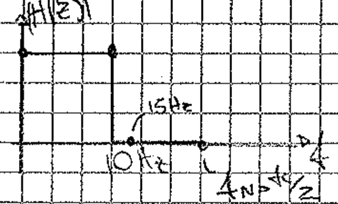
si dà : limite minimo in banda passante
limite massimo in banda attenuata

funzioni di supporto che aiutano a generare l'ordine corretto del filtro (minimo ordine)
↳ buttord, cheb1ord, cheb2ord, ellipord

design diretto: approssimazione diretta del filtro digitale

yulewalk (ordine, vettore freq., vettore ampiezze)

↳ devono essere accoppiati che se li plotto disegno il filtro



$f = [0 \ 10 \ 15 \ 50]/50;$ → il 1° deve essere 0 e l'ultimo f_N

$H = [1 \ 1 \ 0 \ 0];$

[b,a] = yulewalk (9, [0,10,15,50]/50, [1,1,0,0])

FILTRI FIR

- possono avere fase lineare
- sono sempre stabili
- hanno un transitorio di durata finito
- ma sono molto lenti → ordine + grande (molto + alto) → filtro + lunghi
  ↳ introducono + ritardo

MATLAB: b = fir1 (ordine, freq. taglio)

design diretto: b = remez (m, f, m) → come prima yulewalk

x filtri multibanda: b = fir2 → non lo usiamo

I filtri FIR sono caratterizzati da delle gobbe

fase: mi interessa lineare solo nella banda passante

potrebbe esserci uno spostamento di $\pm 2\pi$ → è sempre lineare

Se il vettore dei coefficienti b è SIMMETRICO o ANTISIMMETRICO il filtro è a fase LINEARE

↳ condizione NECESSARIA e SUFFICIENTE: se voglio verificare se è lineare guardo b

(con i comandi di prima matlab si sforza di fare filtri a fase lineare)

RIMOZIONE DELL'INTERFERENZA DI RETE

- Accoppiamento capacitivo con la rete → trasferimento energetico
- sbilanciamento dell'impedenza degli elettrodi (abbassa il CMRR)

CMRR accettabile ≥ 100 dB

il rumore è una sinusoide a circa 50Hz ($\pm$2Hz) di ampiezza variabile

↳ ci sono tecniche che funzionano bene supponendo l'ampiezza di rete costante

es. filtri rigettabanda

FILTRO NOTCH

Funziona bene quando l'ampiezza della sinusoide di rete è costante

Supponiamo di campionare a 250Hz → è un multiplo di 50 $\quad K = \frac{f_c}{f_0} = 5$ (f_0 da rigettare)

Per il filtro NOTCH la f_c deve essere multiplo di 50 → K intero

Ogni periodo della sinusoide di rete è fatto da K=5 campioni → il sesto è all'inizio del ciclo dopo se sottraggo alla sinusoide di rete S campioni prima è zero → però deve essere ad ampiezza costante

$x[n] = s[n] + n[n]$

$x[n-k] = s[n-k] + n[n-k] = s[n-k] + n[n]$ faccio la differenza tra questi due e rumore se ne va

$y[n] = x[n] - x[n-k]$ $Y(z) = X(z) - X(z) \cdot z^{-k} = X(z)(1 - z^{-k})$

$H(z) = \frac{1 - z^{-k}}{1}$ $b = [1\ 0\ 0\ \ldots\ 0\ -1]$ $a = 1$

in un filtro notch l'ordine del filtro è determinato da k. è una struttura a minimo numero di ritardi

Ha fase LINEARE perché b è sempre ANTISIMMETRICO

Però normalmente si campiona a frequenze potenze di 2 (es 256) → k non è un intero

Devo RICAMPIONARE: per il teorema di Nyquist posso ricostruire il segnale analogico (senza perdite) e ricampionarlo ad un multiplo della frequenza di rete

MATLAB: resample

Se invece il rumore non ha ampiezza costante, non so se il SNR aumenterà

slide 31: filtro a PETTINE

$k = 10 \rightarrow f_c = 500\,Hz$

Dato che la sinusoide è periodica, ho un contributo nullo ogni k campioni → ci sono buchi nel modulo a 50 - 100 - 150 ... tutti i multipli

Questo filtro in banda passante ha un guadagno (→ overshoot) (è una cosa che non voglio)

↳ ci sono le oscillazioni a fronte di un k vincolato: il filtro è obbligato a fare queste oscillazioni con un certo ordine: ho un n° limitato di radici (tutte uguali ...)

in continuo: $H(j2\pi f) = \frac{1 - e^{j2\pi fk}}{1}$ → in 0: $H(0) = 0$ la sequenza filtrata è a valor medio nullo

Vantaggi: è FIR → fase lineare (non distorce)

conosco già l'ordine e i coefficienti

è facilissimo da implementare anche ad es. su un ecg (sulla strumentazione)

↳ normalmente sugli ecg c'è un tasto per attivarlo

ha un transitorio di carico (k campioni), poi funziona in real time

[con i filtri FIR a fase lineare, non fare mai il doppio passaggio]

slide 32: il segnale filtrato cambia ma non è distorto → cambia xké ho sottratto le armoniche a 100 Hz ...

FILTRO RICORSIVO

notch che vale 1 a cui sottraggo a 50Hz una delta di dirac

↳ filtro fisicamente non realizzabile: ci accontentiamo di un'approssim.

metto prima un polo e poi uno zero → voglio metterli vicino (ma non uguali) → si annullano

⇒ è + facile generare 2 poli complessi coniugati e 2 zeri.

filtro ricorsivo → 2 poli e 2 zeri complessi coniugati posizionati vicino intorno a 50 Hz

filtro ARMA di ordine 2 2

$b = [1\ c_1\ c_2]$ $a = [1\ c_3\ c_4]$

c_1, c_2, c_3, c_4 si generano come approssimazione digitale della funzione analogica

MATLAB : abbiamo già la funzione che ci chiede in ingresso

- $f_{campionamento} = 50$ Hz
- $T_c = 1/f_c$
- attenuazione a f_{co} (unità lineari)
- larghezza di banda (Hz)
  → a metà dell'altezza

più la banda rigettata è stretta, meno attenua

la fase non è lineare → si può usare il doppio passaggio

funziona anche se l'ampiezza di rete è variabile

FILTRO ADATTIVO

filtro puramente digitale → si basa sulla conoscenza della sinusoide: tento di creare una sinusoide identica al rumore (insegue il rumore) → SOTTRAZIONE
→ creare un'approssimazione il + fedele possibile
Si può usare x rete con ampiezza variabile

Il vincolo è che devo avere una registrazione in cui c'è solo rumore → ovvero dei tratti isoelettrici, in cui c'è solo rumore → es. segnale ecg

Devo inseguire il rumore dove non c'è il segnale

x_n : rumore di rete (segnale che misuro quando c'è solo rumore)

e_n : stima della sinusoide x_n (freq = 50 Hz, ampiezza variabile)

$e_n = A \sin(\omega nT)$ $e_{n+1} = A\sin(\omega nT + \omega T)$ $e_{n-1} = A\sin(\omega nT - \omega T)$

$e_{n+1} = 2A\sin(\omega nT)\cos(\omega nT) - A\sin(\omega nT - nT) = 2\cos(\omega nT) \cdot e_n + e_{n-1}$

→ PREDITTORE: predico sulla base di e_n (attuale) e e_{n-1} (precedente) e_{n+1}

se la predizione è corretta $e_{n+1} = x_{n+1}$

se è scorretta $d_{n+1} = (x_{n+1} + e_{n+1}) - (x_n - e_n)$

Questo filtro tende ad inseguire il rumore: all'inizio sbaglia, ma piano piano tende a diminuire l'errore, la differenza (d_{n+1}) → guardando la differenza attuale, provo ad adattarmi al rumore per il campione dopo

- $e'_{n+1} = e_{n+1} + c$
- $e'_{n+1} = e_{n+1} - c$
- iterativo se $d_{n+1} = 0$

c è una quantità che deve essere scelta in base a quanto tempo ho (ecg → frequenza cardiaca) ↗ scelta dall'utente
→ c alto se ho ~~tanto~~ poco tempo, c basso se ho tanto tempo

questo filtro lavora in condizioni di adattamento quando non ho il segnale, poi si disadatta quando ce l'ho, e continuo così