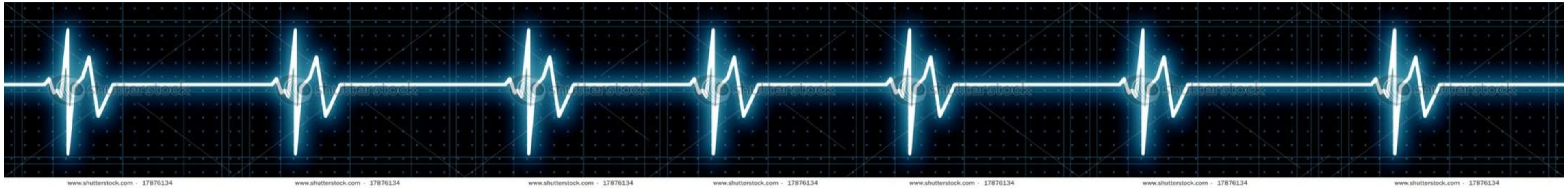
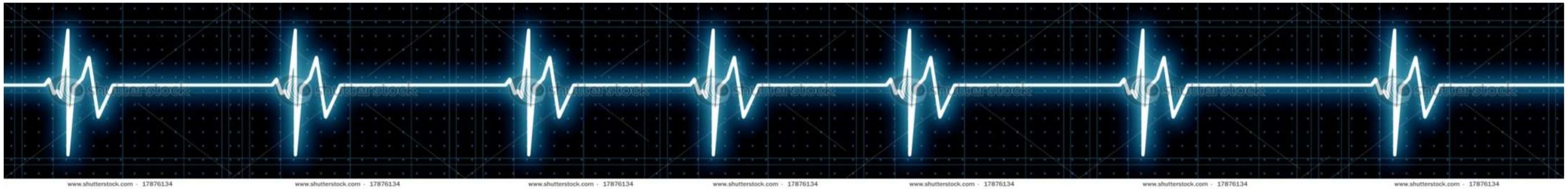


Elaborazione di segnali biologici deterministici: filtri numerici



Come si rappresenta un segnale?

Matematicamente un segnale è rappresentato come funzione di una o più variabili indipendenti.

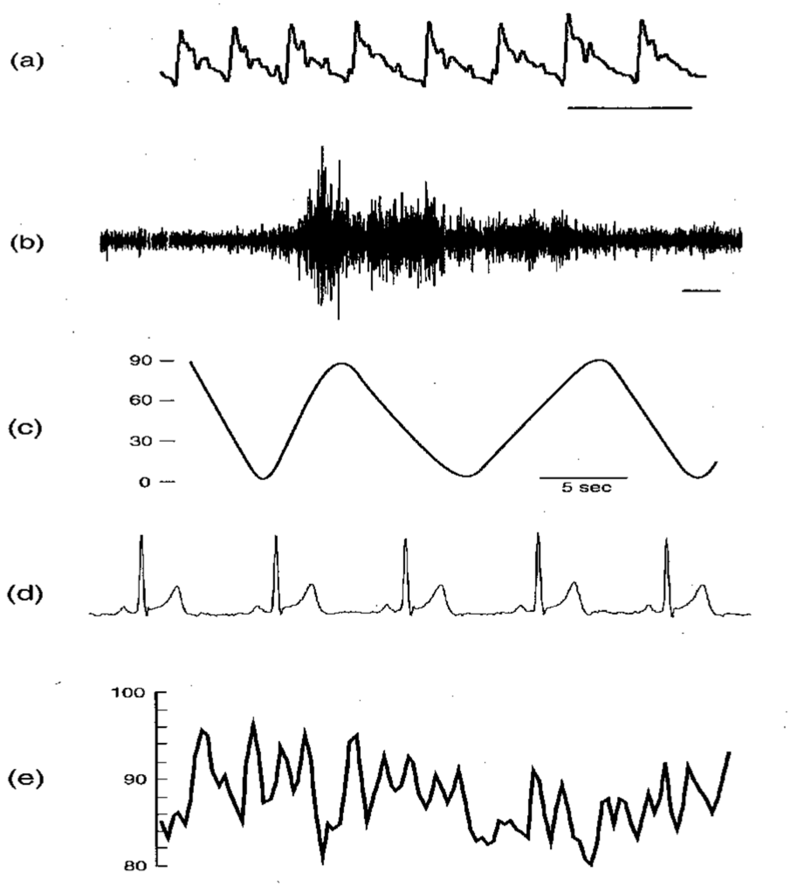
E' convenzione diffusa considerare il tempo come variabile indipendente della rappresentazione matematica di un segnale monodimensionale. Un segnale sarà quindi una funzione

x(t) segnale a tempo continuo

x(n) segnale a tempo discreto

Per i segnali biologici si possono adottare descrizioni e metodi di analisi molto diversi, in quanto…..

…… hanno caratteristiche e forme molto diverse: transitori, periodici, pseudoperiodici…..



a) Velocità del flusso sanguigno nell'arteria cerebrale di un soggetto umano

b) EMG (contrazione e rilassamento della lingua)

c) Angolo di rotazione del ginocchio

d) ECG

e) Frequenza cardiaca istantanea in battiti al minuto (100 battiti)

In particolare, è importante la distinzione tra segnali deterministici e segnali aleatori, perché esistono tecniche di analisi per segnali deterministici e tecniche di analisi per segnali aleatori

Grandezze deterministiche e aleatorie

Grandezze deterministiche:

- Note con precisione
- Rappresentabili in modo univoco
- Una singola osservazione è rappresentativa del fenomeno

Es. età di una persona - N° dei lati di un quadrato - Tensione ai capi di una resistenza di 1 Ohm se la corrente è di 1mA

Grandezze casuali (aleatorie):

- Non assumono valori univoci
- Una singola osservazione non è rappresentativa del fenomeno

Es. età di una persona presa a caso da una popolazione
esito del lancio del dado

Esistono grandezze deterministiche?

la misura quasi sempre comporta un certo grado di approssimazione….

Esistono grandezze aleatorie?

l'aleatorietà è spesso legata al nostro grado di ignoranza del fenomeno……

Useremo allora:

l'approccio deterministico per studiare

- fenomeni "semplici"
- singole osservazioni

l'approccio aleatorio per studiare

- fenomeni "complessi"
- fenomeni che coinvolgono un numero elevato di realizzazioni

Segnali a tempo discreto (deterministici)- richiami

Segnale a tempo discreto x(n) definito per valori interi di n

Spesso è un segnale ottenuto da un segnale continuo per campionamento. In tal caso:

x(t) è il segnale continuo

$$x^*(t) = \sum_{n=-\infty}^{\infty} x(nT) \cdot \delta(t - nT)$$

è il segnale continuo campionato (diverso da zero solo negli istanti di campionamento)

x(n) = x(nT) è il segnale discreto, definito per i soli valori interi di n

Nel passare da segnale campionato a segnale discreto si perde la nozione di tempo di campionamento!!!!

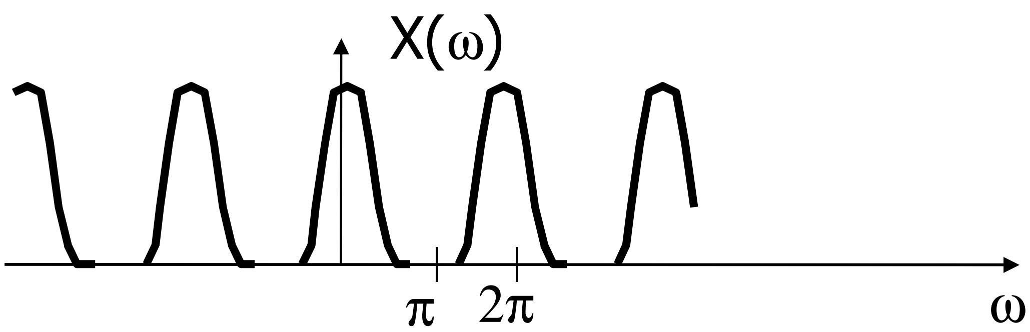
Rappresentazione in frequenza di segnali a tempo discreto

Dato un segnale x(n), la sua trasformata di Fourier continua è così definita:

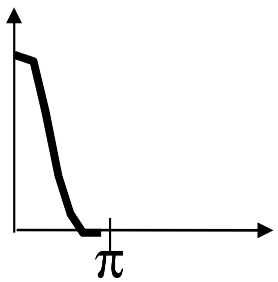
$$X(\omega) = FT\left[x(n)\right] = \sum_{n=-\infty}^{\infty} x(n) e^{-j\omega n}$$

Funzione complessa di variabile reale ω

Periodica di periodo 2π



Se il segnale è reale (praticamente tutti i segnali biomedici lo sono!), la sua FT ha modulo pari e fase dispari



Pertanto è sufficiente rappresentare modulo e fase (o parte reale e imm) nell'intervallo 0 - π

Proprietà della FT

Linearità: $FT[ax_1(n)+bx_2(n)]=aX_1(\omega)+bX_2(\omega)$

Traslazione $FT[x(n+k)]=X(\omega)\, e^{j\omega k}$

Convoluzione y(n)=somma di conv tra x(n) e h(n)

$Y(\omega) = X(\omega)\, H(\omega)$

Trasformata di Fourier Discreta

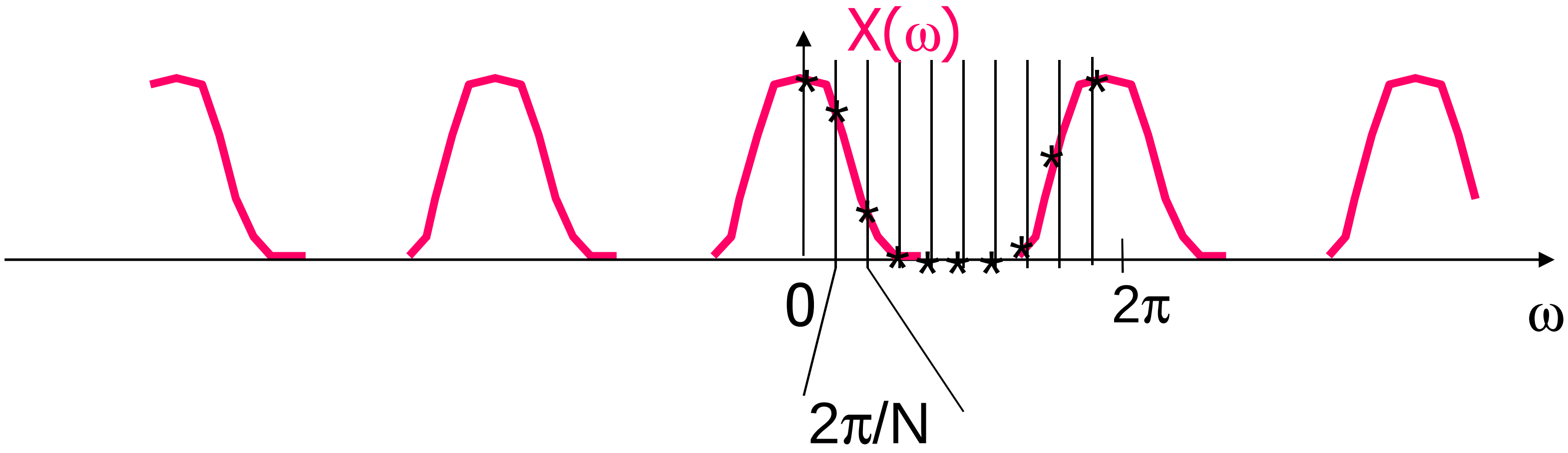
Dato un segnale x(n) con n=0…N-1 (segnale di durata finita), accanto alla trasformata continua di Fourier è possibile definire la Trasformata di Fourier Discreta così definita:

$$X(k) = \mathrm{DFT}[x(n)] = \sum_{n=0}^{N-1} x(n) e^{-j\frac{2\pi}{N}nk}$$

K=0….N-1

E' un segnale complesso ad ascissa discreta, con un numero di campioni pari ad N

Dal confronto tra le formule della FT e della DFT è immediato verificare che X(k) rappresenta la sequenza di campioni equispaziati di un periodo della X(ω) (N campioni nell'intervallo 0-2π)



Se il segnale discreto è la versione campionata di un segnale continuo, qual'è il ruolo della frequenza di campionamento?

FT del segnale campionato

$$X^*(\Omega) = FT\left[x^*(t)\right] = FT\left[\sum_{n=-\infty}^{\infty} x(nT)\delta(t - nT)\right] = \sum_{n=-\infty}^{\infty} x(nT)e^{-j\Omega nT}$$

FT della sequenza

$$X(\omega) = FT\left[x(n)\right] = \sum_{n=-\infty}^{\infty} x(n)e^{-j\omega n}$$

Essendo $x(n) = x(nT)$, si ha $X(\omega) = X^*(\Omega)$ con $\omega = \Omega T$
quindi rappresentiamo la FT della sequenza:

in termini di pulsazione ω da 0 a π
in termini di frequenza f da 0 a 0.5

che corrispondono, nel continuo, ai seguenti intervalli
in termini di pulsazione Ω da 0 a $0.5\Omega_C = \pi F_c$
in termini di frequenza F da 0 a 0.5Fc
MATLAB da 0 a 1

ω viene indicata come pulsazione normalizzata $\omega = \Omega T = \Omega/F_c$
f come frequenza normalizzata f=F/Fc

Trasformata zeta

Dato un segnale x(n), la sua trasformata zeta è così definita:

$$X(z) = ZT\left[x(n)\right] = \sum_{n=-\infty}^{\infty} x(n) z^{-n}$$

Funzione complessa di variabile complessa z

Non cercheremo di rappresentare il grafico di X(z). Vedremo che essa è utile per rappresentare segnali e sistemi a tempo discreto.
Per ogni sequenza l'insieme di valori per cui la trasformata Z converge viene detta regione di convergenza. Generalmente si intende la convergenza assoluta:

$$\exists M > 0 : \sum_{n=-\infty}^{\infty} \left| x(n) z^{-n} \right| < M$$

Tale condizione coinvolge il modulo di z. Pertanto le regioni di convergenza sono dei settori circolari $R^- < |z| < R^+$

ESEMPI	**ZT**	**regione conv**
Campione unitario $\delta(n)$	**1**	**tutto il piano z**
Gradino unitario u(n)	**z/(z-1)**	**$\|z\|>1$**
Esponenziale $a^n u(n)$	**z/(z-a)**	**$\|z\|>\|a\|$**
Segnale costante x(n)=1 per ogni n		**nessun valore di z**
Segnale causale		**$\|z\|>R^-$**

Proprietà della ZT

Linearità: $ZT[ax_1(n)+bx_2(n)]=aX_1(z)+bX_2(z)$

Traslazione $ZT[x(n+k)]=X(z)\, z^k$

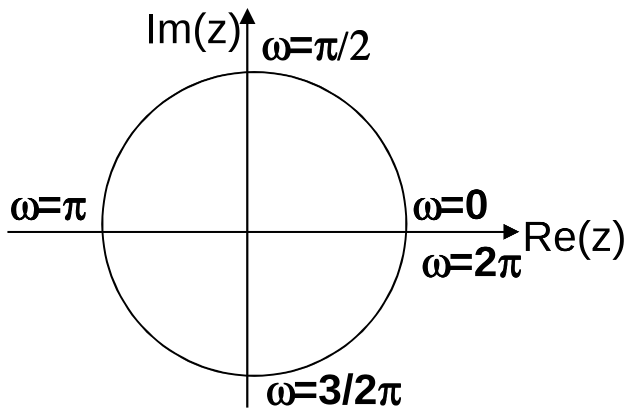
Convoluzione y(n)=somma di conv tra x(n) e h(n)
$Y(z) = X(z)\, H(z)$

Relazione tra la ZT e la FT

Confrontando le definizioni:
$$X(z) = ZT\left[x(n)\right] = \sum_{n=-\infty}^{\infty} x(n) z^{-n}$$
$$X(\omega) = FT\left[x(n)\right] = \sum_{n=-\infty}^{\infty} x(n) e^{-j\omega n}$$

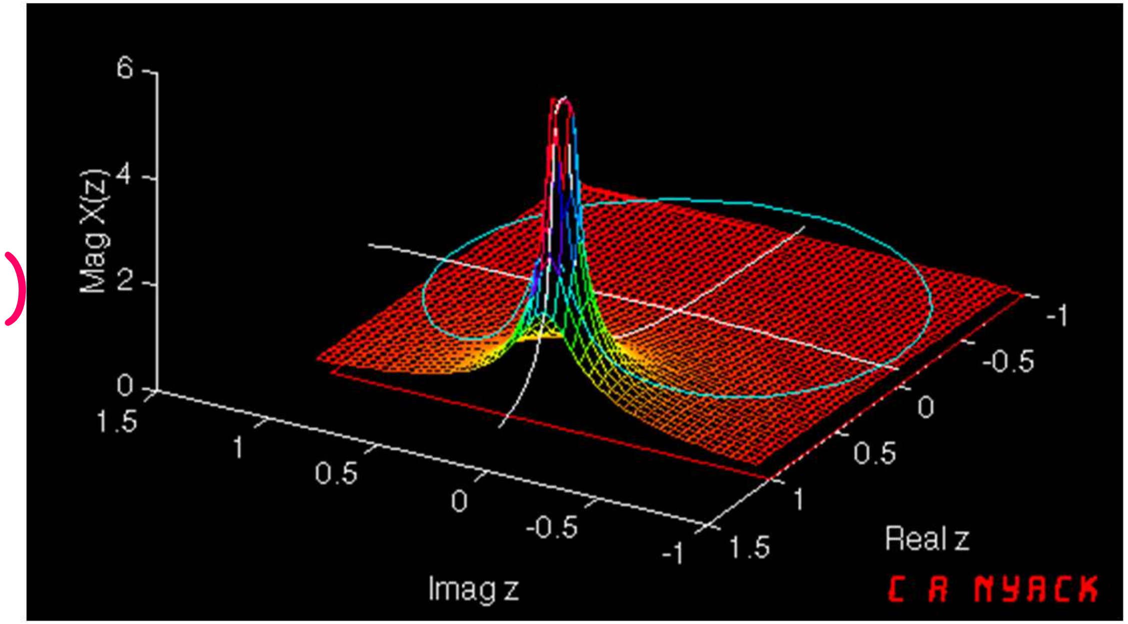
Si ottiene la FT valutando la ZT sul cerchio di raggio unitario, $z=e^{j\omega}$

$X(\omega)=X(z)|^{z=\exp(j\omega)}$

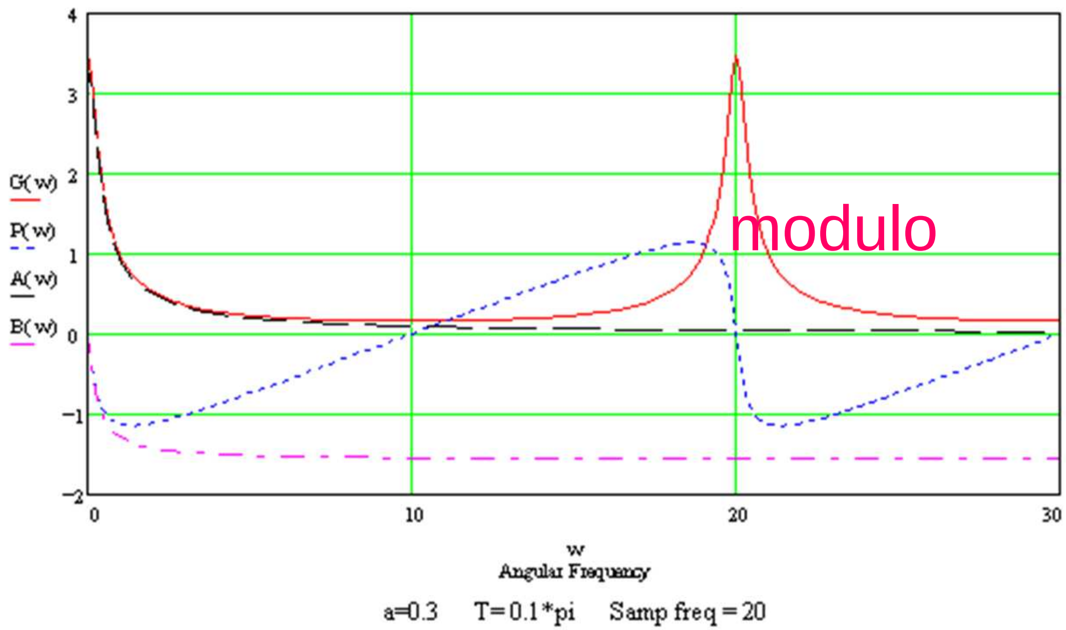


Relazione tra ZT e FT ESEMPIO

ZT(modulo)



FT



Sistema a tempo discreto - richiami

Trasformazione che, in corrispondenza ad un certo segnale di ingresso x(n) produce un segnale di uscita y(n)

$$x(n) \longrightarrow \boxed{\text{Sistema}} \longrightarrow y(n)$$

Sistema Lineari Invarianti alla Traslazione

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

MATLAB
y=conv(x,h)

h(n) è la risposta del sistema al campione unitario δ(n)

descrive completamente il sistema

$$\delta(n) \longrightarrow \boxed{\text{SLIT}} \longrightarrow h(n)$$

Sistemi FIR

h(n) di durata finita

Sistemi IIR

h(n) di durata non finita

Proprio perché h(n) riassume in se tutte le caratteristiche del sistema LIT, è possibile testare su h(n) le proprietà di stabilità e causalità

Stabilità (BIBO) di sistemi LIT

Garantisce che la risposta del sistema non diverga, quando l'ingresso è limitato

Condizione necessaria e sufficiente:

$$\exists M : \sum_{n=-\infty}^{\infty} |h(n)| < M$$

Causalità di sistemi LIT

Garantisce la possibilità di elaborare il segnale in linea

Condizione necessaria e sufficiente:

$$h(n) = 0 \quad n < 0$$

ESEMPIO 1

h(n) = 1 n=0,1,2,3,4

= 0 altrove

Filtro FIR

È stabile e causale

ESEMPIO 2

h(n) = $a^n u(n)$

Filtro IIR

E' causale

E' stabile se |a|<1

Esiste una sottoclasse di SLIT in cui il legame I/O è esprimibile come equazione alle differenze

$$\sum_{k=0}^{N} a_k y(n-k) = \sum_{k=0}^{M} b_k x(n-k)$$

Gli ordini N e M sono finiti

L'equazione alle differenze serve come realizzazione numerica del sistema

Forma causale:

$$y(n) = -\underbrace{\sum_{k=1}^{N} a_k y(n-k)}_{\text{Parte AR}} + \underbrace{\sum_{k=0}^{M} b_k x(n-k)}_{\text{Parte MA}}$$

MATLAB
$b=[b_0\ b_1....b_M]$
$a=[1\ \ a_1....a_N]$
y=filter(b,a,x)

Sistemi FIR

h(n) di durata finita
Possono essere descritti da una equazione alle diff <u>non ricorsiva</u>, cioè con la sola parte MA

Sistemi IIR

h(n) di durata non finita
Devono essere descritti da eq. alle differenze <u>ricorsive</u>, cioè con parte AR e MA

ESEMPIO 1

h(n) = 1 n=0,1,2,3,4
= 0 altrove
Filtro FIR
La somma di convoluzione fornisce l'eq. alle diff:
y(n)=x(n)+x(n-1)+x(n-2)+x(n-3)+x(n-4)
che ha solo parte MA

esiste però anche una eq alle diff MA+AR che realizza il sistema, cioè ha la stessa h(n)
y(n)=y(n-1)+x(n)-x(n-5)

ESEMPIO 2

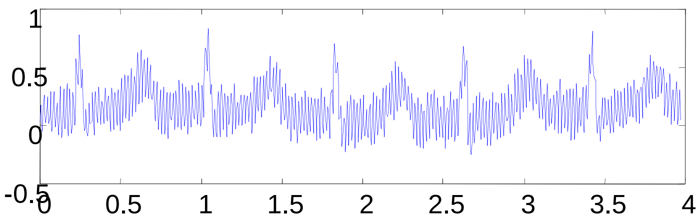
h(n) = a^nu(n)
Filtro IIR
La somma di convoluzione NON fornisce l'eq alle diff perché la sommatoria ha un numero infinito di termini

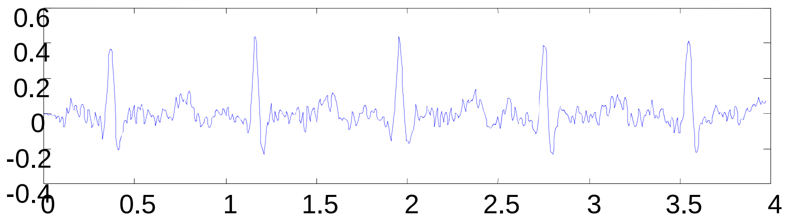
L'eq. alle diff che realizza il sistema è
y(n) = ay(n-1)+x(n)

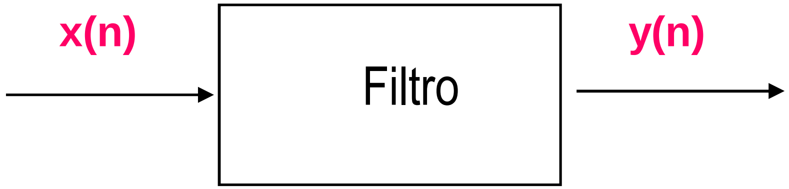
Un sistema a tempo discreto lineare, invariante alla traslazione, stabile è un Filtro Numerico

Viene applicato ad un segnale a tempo discreto per modificare il suo contenuto in frequenza, ad es. per migliorare il SNR, se segnale e rumore occupano bande diverse

ECG + rumore







E’ pertanto importante rappresentare segnali e sistemi nel dominio della frequenza

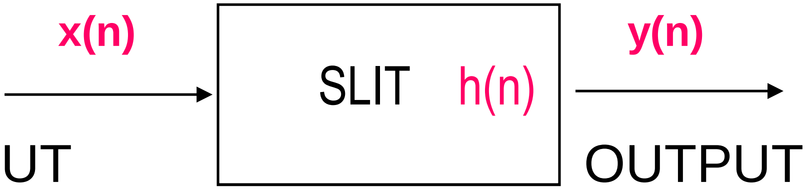
Rappresentazione FT di sistemi LIT a tempo discreto

Passando nel dominio della frequenza:

$$Y(\omega)=H(\omega)X(\omega)$$

con $H(\omega)=FT[h(n)]$

$H(\omega)$ è la risposta in frequenza del sistema. Perché si chiama così?
Applichiamo al sistema il segnale sinusoidale $x(n) = A\,\mathrm{sen}(\omega_0 n + \phi)$



INPUT
Ampiezza A
Pulsazione ω_0
Fase ϕ

OUTPUT
Ampiezza A $|H(\omega_0)|$
Pulsazione ω_0
Fase ϕ +arg $H(\omega_0)$

$|H(\omega_0)|$ è il fattore per cui viene modificata l'ampiezza del segnale sinusoidale di pulsazione ω_0, arg $H(\omega_0)$ è la variazione della sua fase iniziale

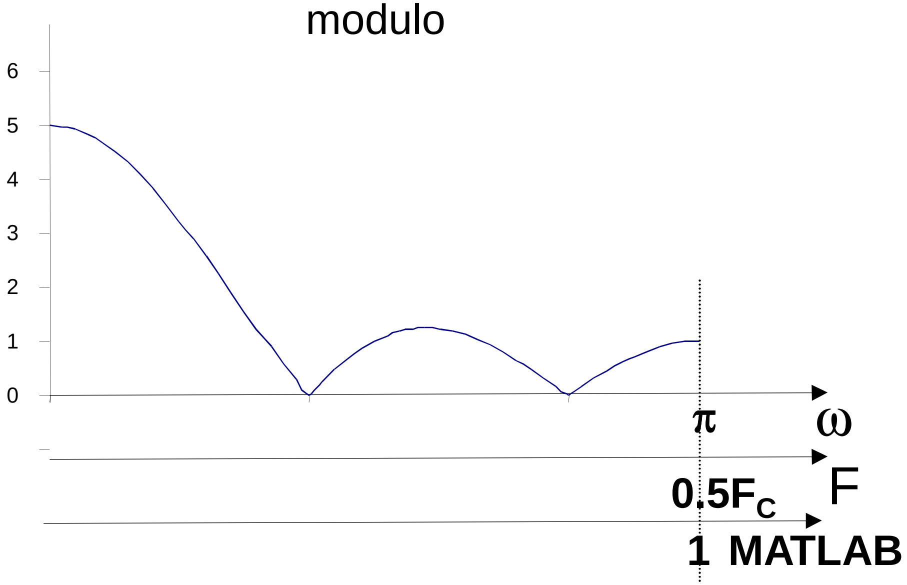
ESEMPIO 1

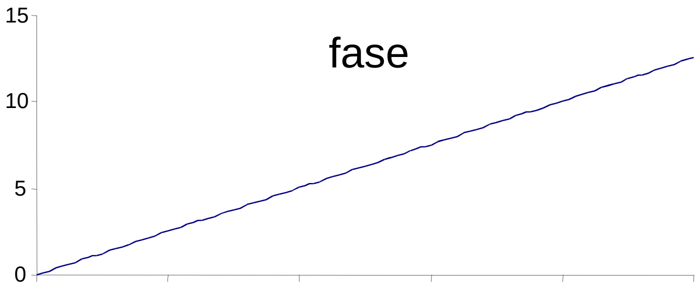
y(n)=x(n)+x(n-1)+x(n-2)+x(n-3)+x(n-4)

h(n)=1 n=0,1,2,3,4

0 altrove

$$H(\omega) = e^{-j2\omega} \frac{\mathrm{sen}(5\omega/2)}{\mathrm{sen}(\omega/2)}$$





Il modulo si annulla ad es. a $2\pi/5$

Quindi se $x(n) = \cos(0.4\pi n)$

$y(n) = 0$

Verifica:

x(n)

1
.31
-0.81
-0.81
0.31
1
0.31
-0.81
-0.81
0.31
1 ecc.

y(n) = 0 perché sommando 5 campioni consecutivi si ottiene sempre zero

Se invece x(n) = cost

y(n) = cost*5

MATLAB

[h,w] = freqz(b,a,n)

Se la fase di H è nulla :

Arg $H(\omega)=0$ per tutte le frequenze, ovvero $H(\omega)$ è reale
il filtro modifica solo il modulo del segnale ma non la sua fase

Quando si verifica?

h(n) è una sequenza pari: $h(n)=h(-n)$ (sistema non causale!!)

Se i la fase di H è lineare:

Arg $H(\omega)=k\omega$, Il filtro introduce uno shift nel segnale:

$$Y(\omega) =| X(\omega)\ H(\omega)|\ \mathrm{Arg}\ X(\omega)\ e^{j\omega k}$$

senza alterare le relazioni temporale tra le sue varie componenti

Quando si verifica?

h(n) è una sequenza simmetrica: $h(N+n)=h(N-n)$

Un sistema causale può avere la fase nulla?

NO

Un sistema causale può avere la fase lineare?

Si, ma solo se h(n) simmetrica e di durata finita (FIR)

ESEMPIO 3

h(n)=3 n=0
2 n=+1, -1
1 n=+2, -2
0 altrove
$H(\omega)=3+4\cos\omega+2\cos 2\omega$
Reale , quindi fase nulla,
perché h(n) pari

ESEMPIO 4

h(n)=1 n=0, 4
2 n=1,3
3 n=2
0 altrove
$H(\omega)=(4\cos\omega+2\cos 2\omega+3)e^{-j2\omega}$
Fase lineare perché h(n) simmetrica
Causale perché h(n)=0 n<0

Rappresentazione ZT di sistemi LIT a tempo discreto

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

con $H(z)=ZT[h(n)]$

H(z) è la funzione di trasferimento del sistema

ESEMPIO 1

$h(n) = 1$ $n=0,1,2,3,4$

$= 0$ altrove

$H(z) = (z^4+z^3+z^2+z+1)/(z^4)$ reg conv: tutto il piano tranne l'origine

ESEMPIO 2

$h(n) = a^n u(n)$ con $|a|<1$

$H(z) = z/(z-a)$ reg conv $|z|>|a|$

Stabilità di sistemi LIT

Condizione necessaria e sufficiente:

$$\exists M : \sum_{n=-\infty}^{\infty} |h(n)| < M$$

La regione di convergenza di H(z) contiene il cerchio di raggio unitario |z|=1

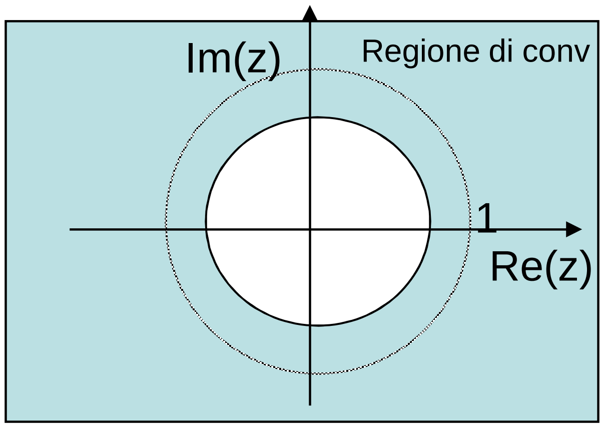
Causalità di sistemi LIT

Condizione necessaria e sufficiente:

$$h(n) = 0 \qquad n < 0$$

La regione di convergenza è la zona esterna ad un cerchio

Stabilità + causalità di sistemi LIT



Sistemi LIT descritti da un'equazione alle differenze

$$\sum_{k=0}^{N} a_k y(n-k) = \sum_{k=0}^{M} b_k u(n-k)$$

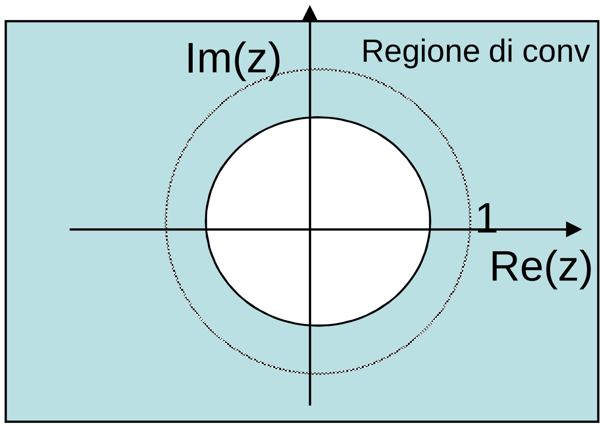
H(z) è un rapporto di polinomi in zeta

$$H(z) = \frac{\sum_{k=0}^{M} b_k z^{-k}}{\sum_{k=0}^{N} a_k z^{-k}}$$

Poli sono i valori di z per cui il denominatore di H(z) si annulla (esterni alla regione di convergenza!)
Zeri sono i valori di z per cui il numeratore di H(z) si annulla
Poli e zeri sono reali o complessi coniugati

MATLAB
poli = roots(a)
zeri = roots(b)

Stabilità + causalità



Tutti i poli devono stare dentro il cerchio di raggio unitario, quindi avere modulo < 1

Ricapitolando:

Sistemi FIR

h(n) di durata finita
possono essere descritti una eq.alle diff non ricorsiva (MA)
Poli solo nell'origine
Sempre stabili

ESEMPIO 1

$y(n)=x(n)+x(n-1)+x(n-2)+x(n-3)+x(n-4)$
$H(z)= (z^4+z^3+z^2+z+1)/(z^4)$
Poli nell'origine FIR
Stabile

ESEMPIO 1

$y(n)=y(n-1)+x(n)-x(n-5)$

Sistemi IIR

h(n) di durata non finita
Devono essere descritti da eq. alle differenze ricorsive (AR+MA)
Almeno 1 polo fuori dall'origine
Stabilità????

ESEMPIO 2

$y(n)=ay(n-1)+x(n)$
$H(z)=z/(z-a)$
Polo in z=a IIR
Stabile se $|a|<1$

ESEMPIO 5

$y(n)=3y(n-1)-2y(n-2)+x(n-1)-3x(n-2)$

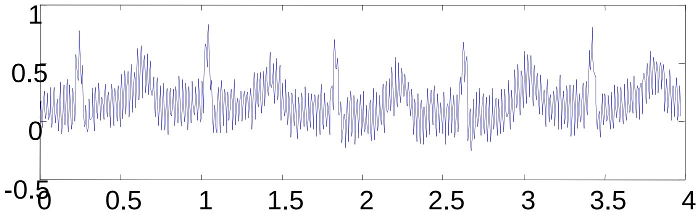
ESEMPIO 6

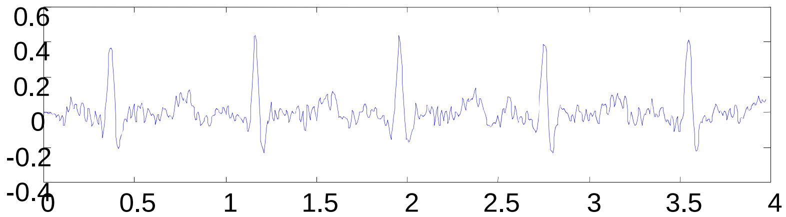
$y(n)=y(n-1)-0.74y(n-2)+x(n)-x(n-1)$

Filtri Numerici

Un sistema a tempo discreto lineare, invariante alla traslazione, stabile è un filtro numerico.

ECG + rumore





x(n) → Filtro → **y(n)**

Viene rappresentato da:

- Equazione alle differenze
- Risposta in frequenza
- Trasformata zeta
- Diagramma zeri-poli nel piano complesso

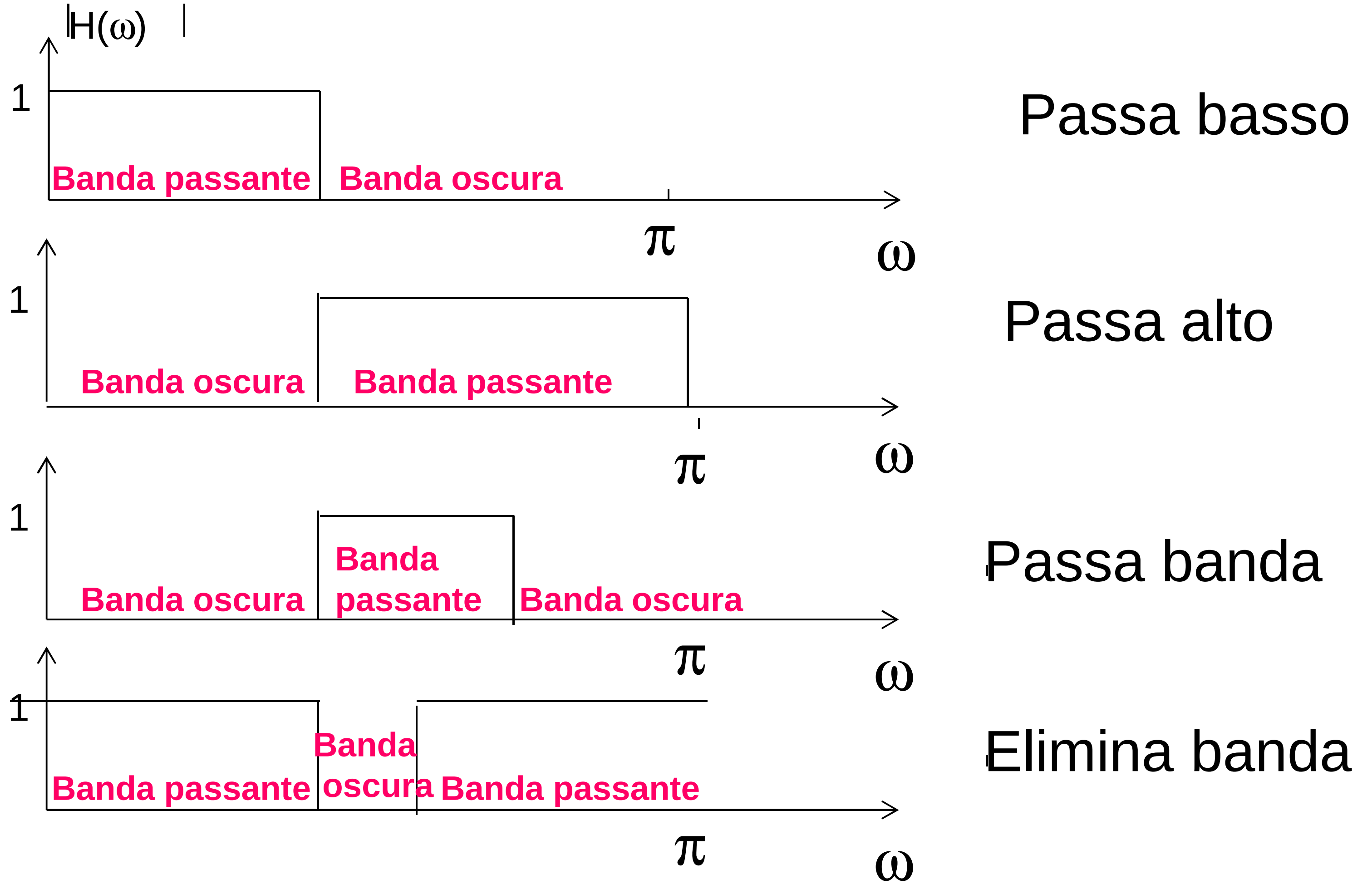
MATLAB
$b=[b_0\ b_1 \ldots b_M]$
$a=[1\ \ a_1 \ldots a_N]$
y=filter(b,a,x)

MATLAB
[h,w] = freqz(b,a,n)

MATLAB
poli = roots(a)
zeri = roots(b)

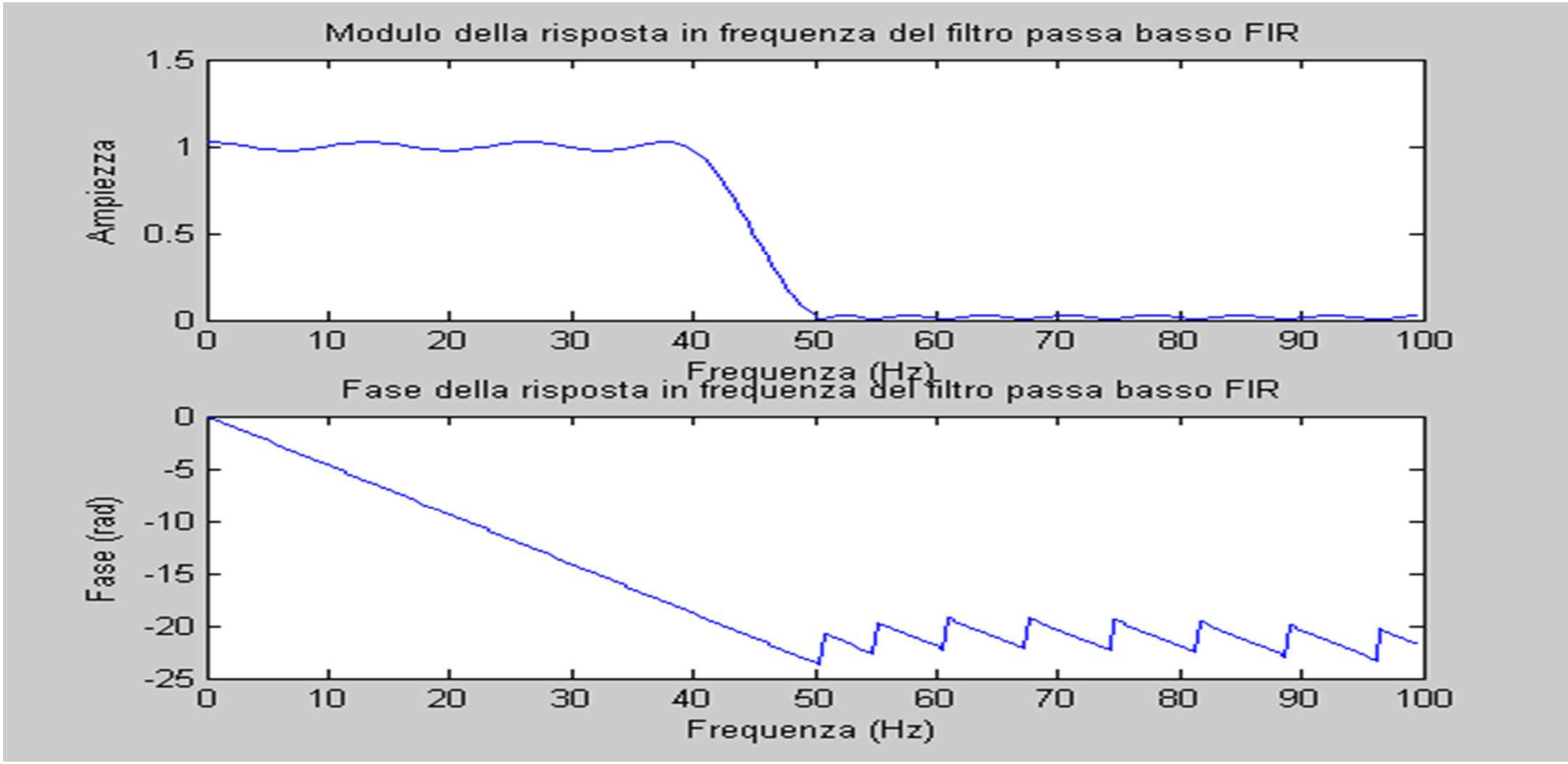
Filtri ideali

Sono definiti in base al modulo della loro risposta in frequenza. Si evidenzia quali frequenze si vogliono far passare inalterate (banda passante) e quali si vogliono eliminare (banda oscura).



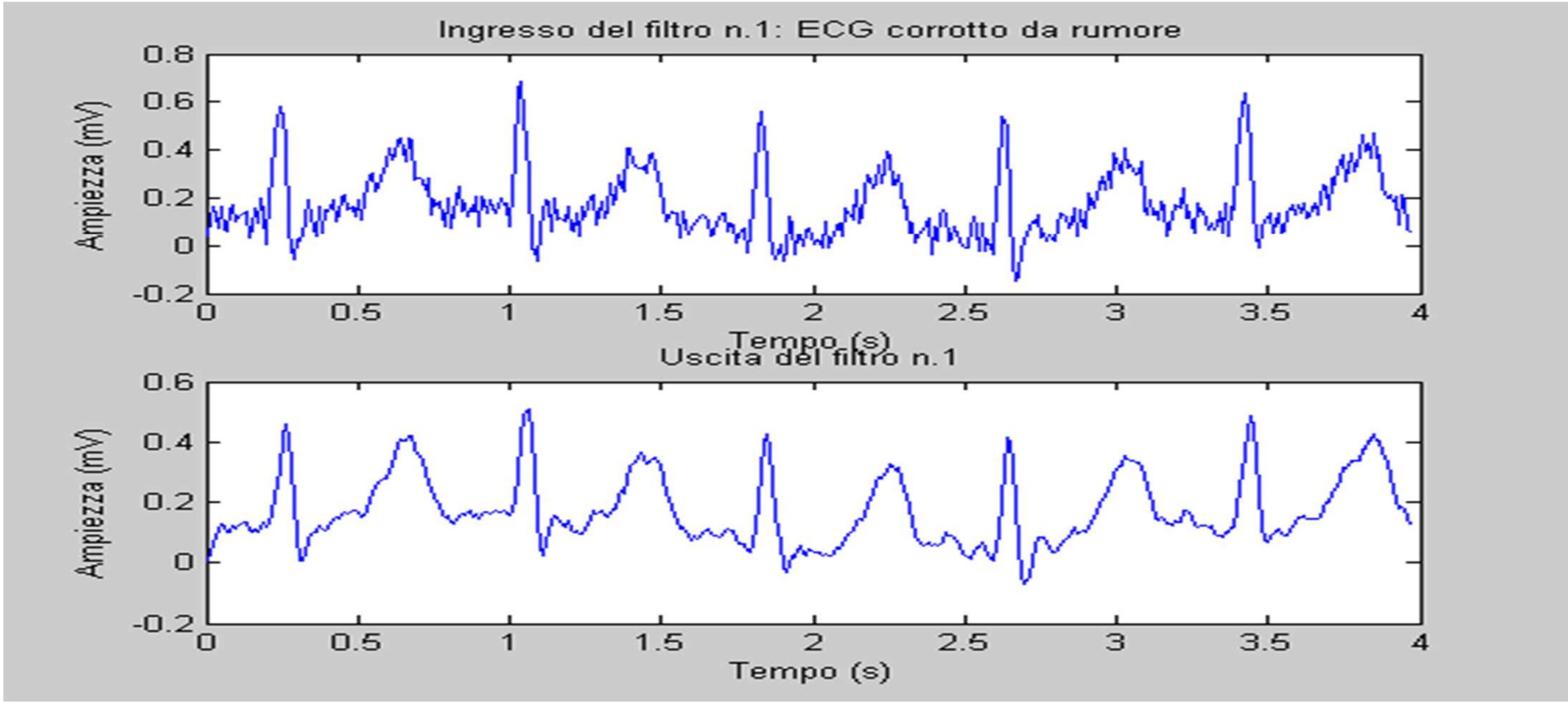
Filtri reali

Nella pratica, si riesce ad ottenere solo una approssimazione del comportamento ideale. Tale approssimazione è tanto più buona, quanto più elevato è l'ordine del filtro (legato al numero di termini che compaiono nell'equazione alle differenze)



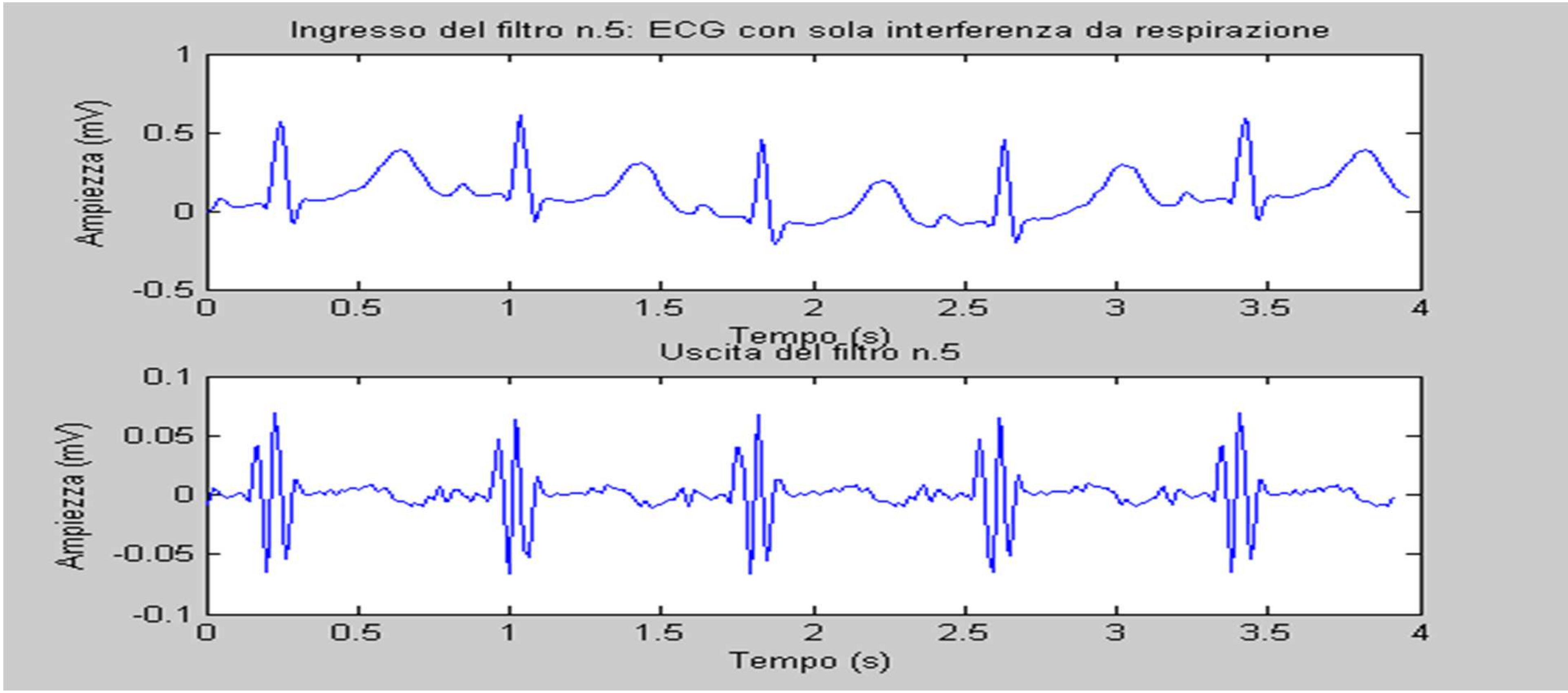
Esempio 1: Filtro Passabasso

Banda passante 0-20Hz
Fc = 100 HZ



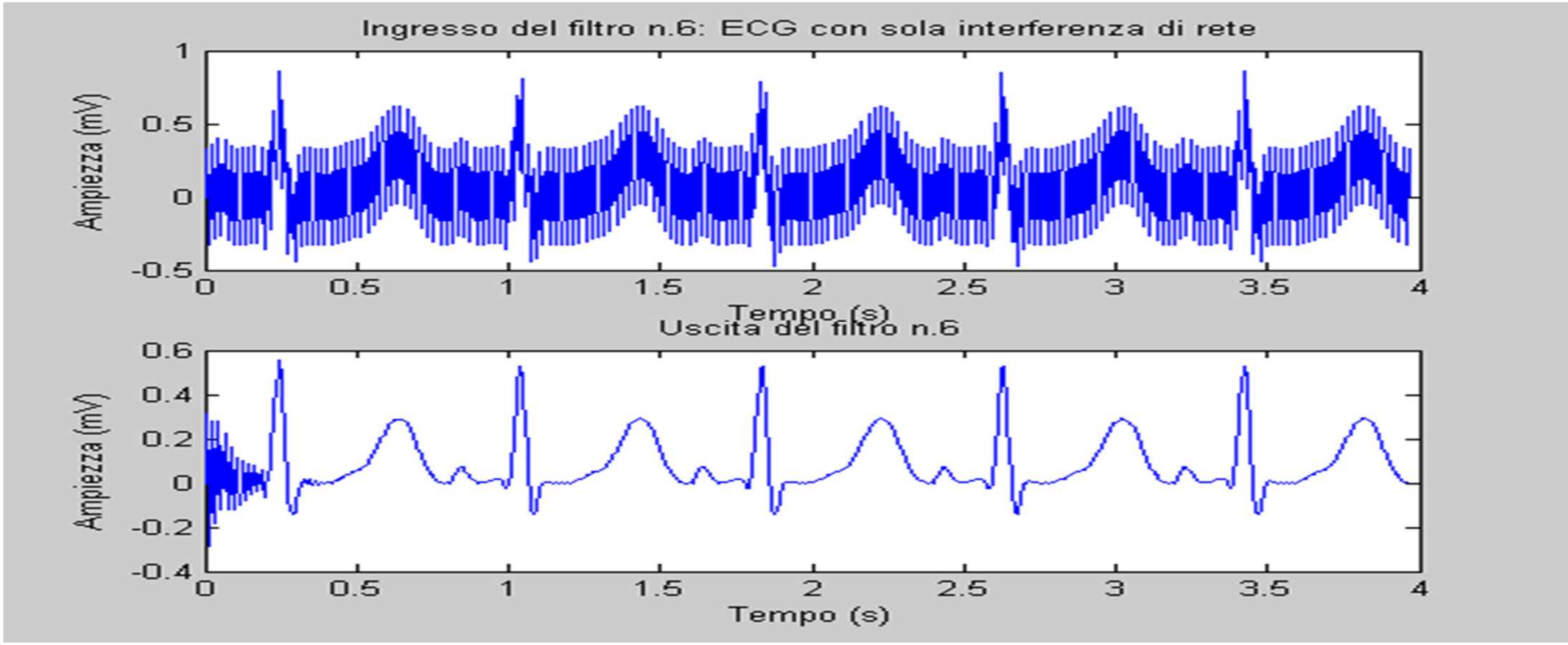
Esempio 2: Filtro Passabanda

Banda passante 7-42 Hz
Fc = 100 HZ



Esempio 3: Filtro Eliminabanda (notch)

elimina 50 Hz
Fc = 100 HZ



Specifiche

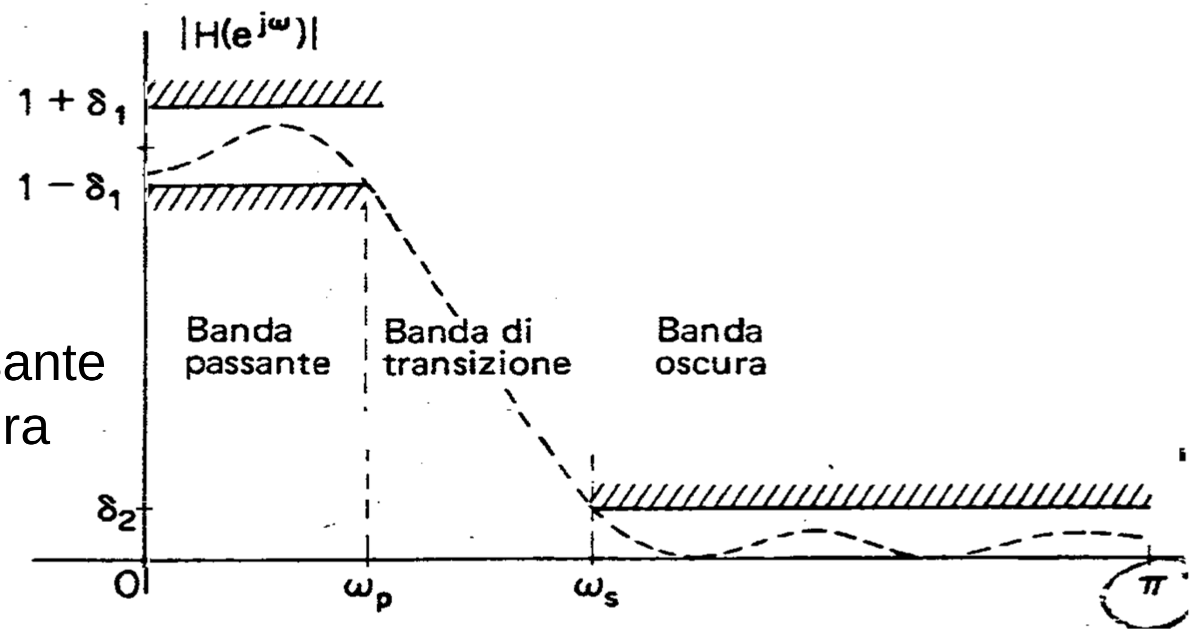
Esempio: filtro PassaBasso

Ideale : $|H(\omega)| = 1 \qquad 0<\omega<\omega_p$

$\qquad\qquad\quad = 0 \qquad$ altrove

Reale: $1-\delta_1<|H(\omega)|<1+\delta_1 \quad 0<\omega<\omega_p$

$\qquad\qquad |H(\omega)|<\delta_2 \qquad \omega>\omega_s$



$\omega_s-\omega_p$ = banda di transizione

δ_1 = ripple in banda passante

δ_2 = ripple in banda oscura

Le specifiche riguardano il modulo. E la fase?

Situazione ideale : fase H(ω)=0 per ogni ω
(il filtro modifica solo il modulo del segnale e non la sua fase)

- Quando si verifica?
  h(n) pari

  quindi filtro non causale

Situazione ancora favorevole : fase H(ω) lineare in ω
(il filtro introduce uno shift nel segnale, senza alterare le relazioni temporale tra le varie componenti del segnale)

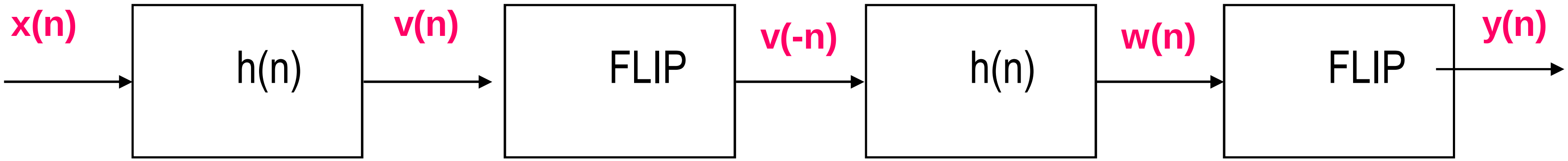
- Quando si verifica?
  h(n) di durata finita (FIR) e simmetrica

  quindi filtro FIR

(condizione necessaria ma non sufficiente!)

Filtraggio Forward-Backward

E' possibile rendere un filtro a fase nulla rinunciando alla causalità, ed implementandolo offline due volte, secondo questo schema



$v(n) = h(n) \otimes x(n)$
$w(n) = h(n) \otimes v(-n) = h(n) \otimes h(-n) \otimes x(-n)$
$y(n) = h(-n) \otimes h(n) \otimes x(n)$

MATLAB
$b=[b_0\ b_1 \ldots. b_M]$
$a=[1\ \ a_1 \ldots. a_N]$
y=filtfilt(b,a,x)

In frequenza:
$Y(\omega) = H(\omega)\, H(-\omega)\, X(\omega) = |H(\omega)|^2\, X(\omega)$
Quindi il filtraggio F-B eleva al quadrato il modulo della risposta in frequenza del filtro H ed annulla la sua fase

Progetto di filtri

Date le specifiche, trovare il filtro (equazione alle differenze) che approssima al meglio tale specifiche.

Vedremo due metodi di sintesi al calcolatore (MATLAB):

Filtri IIR ellittici
Filtri FIR Parks McClellan

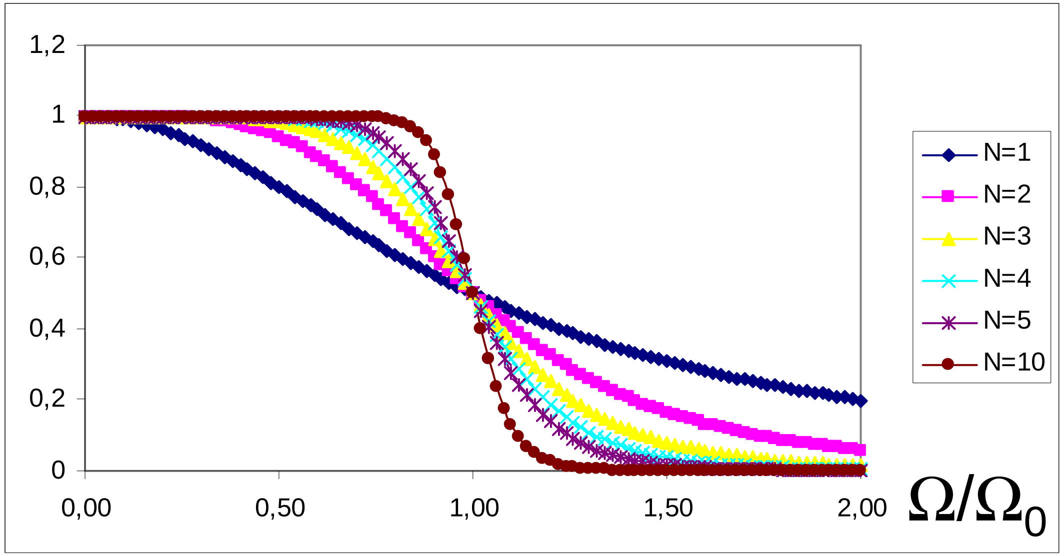
Filtri IIR - sintesi con MATLAB

Si usano dei prototipi, i cui parametri vengono adattati per soddisfare alle specifiche
Sono ottenuti da filtri analogici (piano s) per trasformazione (dal piano s al piano z)

Esistono varie categorie di filtri:
Chebycev
Butterworth
Ellittici

Es. filtri di Butterworth

$$|H(\Omega)|^2=1/(1+(\Omega/\Omega_0)^{2N})$$



Filtri Ellittici

Consentono di realizzano un insieme di specifiche (banda transizione, ripple in banda passante ed oscura) con il valore minimo dell'ordine

2 funzioni MATLAB:

ELLIPORD   date le specifiche (es. per filtro passabasso banda passante, banda di transizione, ripple in banda passante, ripple in banda oscura) valuta l'ordine

ELLIP Date le specifiche e l'ordine, calcola il filtro che soddisfa tali requisiti

Ellipord

ELLIPORD Elliptic filter order selection.

[N, Wn] = ELLIPORD(Wp, Ws, Rp, Rs) returns the order N of the lowest order digital elliptic filter that loses no more than Rp dB in the passband and has at least Rs dB of attenuation in the stopband.

Wp and Ws are the passband and stopband edge frequencies, normalized from 0 to 1 (where 1 corresponds to pi radians/sample). For example,

Lowpass: Wp = .1, Ws = .2
Highpass: Wp = .2, Ws = .1
Bandpass: Wp = [.2 .7], Ws = [.1 .8]
Bandstop: Wp = [.1 .8], Ws = [.2 .7]

ELLIPORD also returns Wn, the elliptic natural frequency to use with ELLIP to achieve the specifications

NOTE: If Rs is much much greater than Rp, or Wp and Ws are very close, the estimated order can be infinite due to limitations of numerical precision.

ELLIP

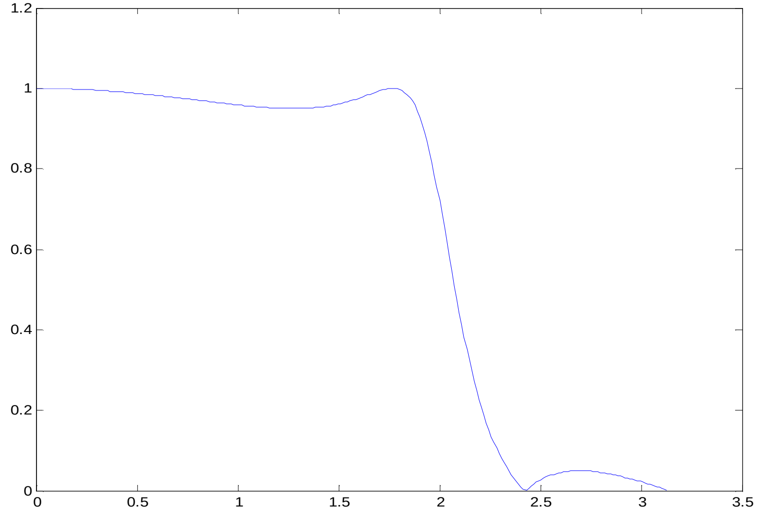
ELLIP Elliptic or Cauer digital and analog filter design.

[B,A] = ELLIP(N,Rp,Rs,Wn) designs an Nth order lowpass digital elliptic filter with Rp decibels of ripple in the passband and a stopband Rs decibels down. ELLIP returns the filter coefficients in length N+1 vectors B (numerator) and A (denominator). The cutoff frequency Wn must be 0.0 < Wn < 1.0, with 1.0 corresponding to half the sample rate. Use Rp = 0.5 and Rs = 20 as starting points, if you are unsure about choosing them.

If Wn is a two-element vector, Wn = [W1 W2], ELLIP returns an order 2N bandpass filter with passband W1 < W < W2.
[B,A] = ELLIP(N,Rp,Rs,Wn,'high') designs a highpass filter.
[B,A] = ELLIP(N,Rp,Rs,Wn,'stop') is a bandstop filter if Wn = [W1 W2].

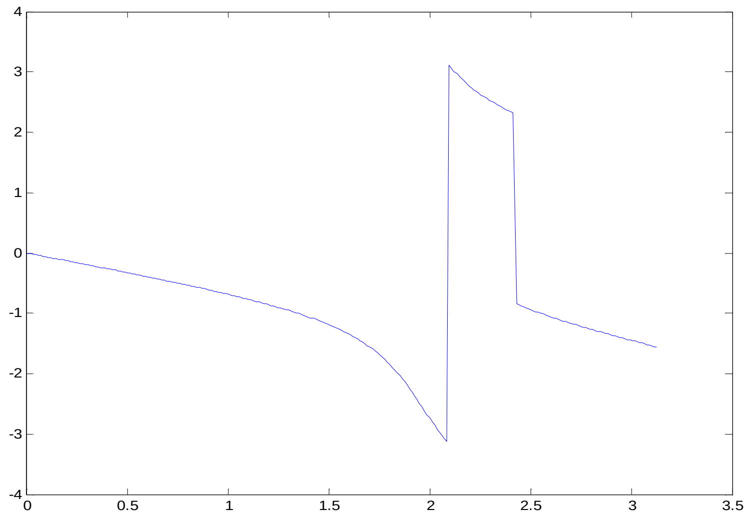
Esempio

FC = 8kHz
Filtro passabasso, banda passante 0-1.6kHz 1-0.95
banda oscura 2.4-4kHz 0-0.05



File MATLAB

```
RpdB=-20*log10(0.95)
RsdB=-20*log10(0.05)
Wp=1600/4000
Ws=2400/4000
[N,wn]=ellipord(wp,ws,RpdB,RsdB)
[b,a]=ellip(N,RpdB,RsdB,wn)
```



Risultati
N = 3, wn=0.6
b = 0.3275 0.8217 0.8217 0.327
a = 1.0000 0.6391 0.6489 0.0105

Se diminuisce la banda di transizione:

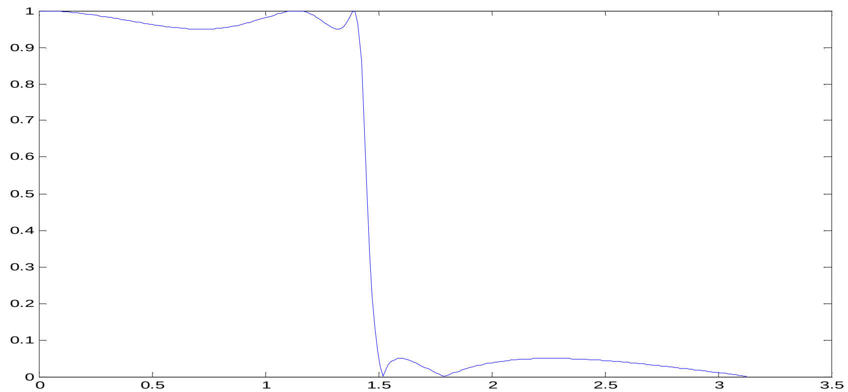
FC = 8kHz

Filtro passabasso, banda passante 0-1.6kHz 1-0.95

banda oscura 1.8-4kHz 0-0.05

File MATLAB

```
RpdB=-20*log10(0.95)
RsdB=-20*log10(0.05)
Wp=1600/4000
Ws=1800/4000
[N,wn]=ellipord(wp,ws,RpdB,RsdB)
[b,a]=ellip(N,RpdB,RsdB,wn)
```

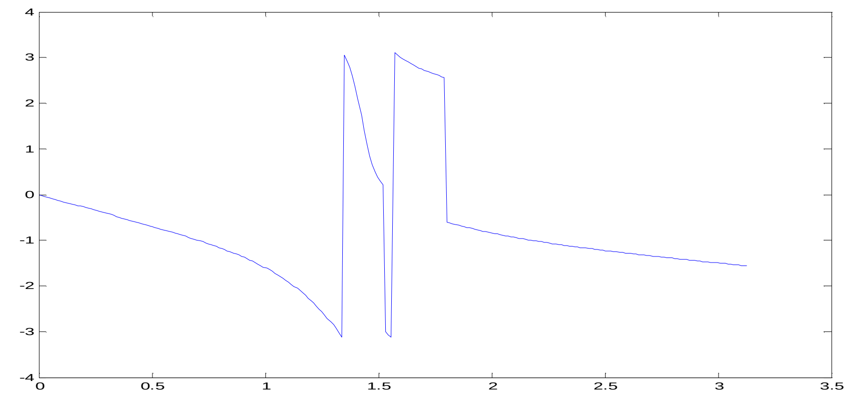


Risultati

N = 5, wn=0.45

b = 0.1387 0.1861 0.3189 0.3189 0.1861 0.1387

a = 1.0000 -1.0209 1.8387 -1.0668 0.7055 -0.1692



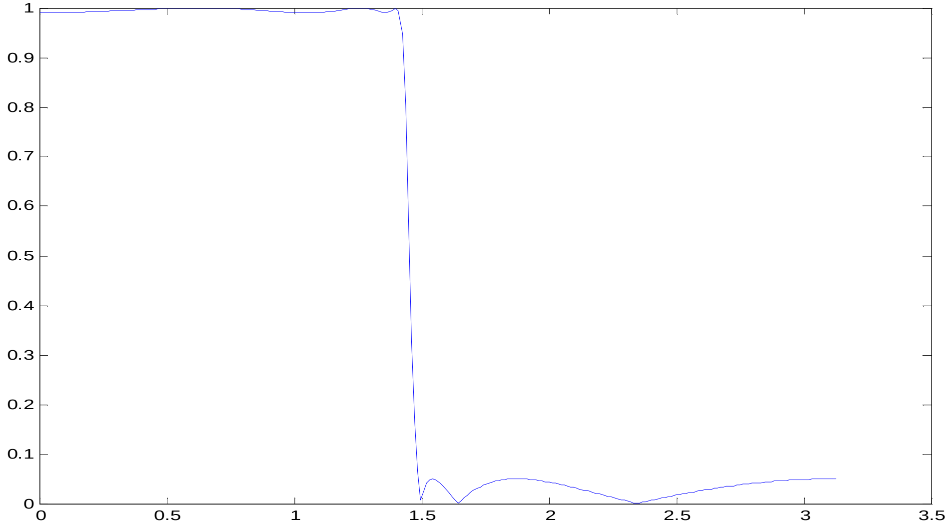
Aumenta l'ordine

Se diminuisce anche il ripple in banda passante

FC = 8kHz
Filtro passabasso, banda passante 0-1.6kHz 1-0.99
banda oscura 1.8-4kHz 0-0.05

File MATLAB

```
RpdB=-20*log10(0.99)
RsdB=-20*log10(0.05)
Wp=1600/4000
Ws=1800/4000
[N,wn]=ellipord(wp,ws,RpdB,RsdB)
[b,a]=ellip(N,RpdB,RsdB,wn)
```

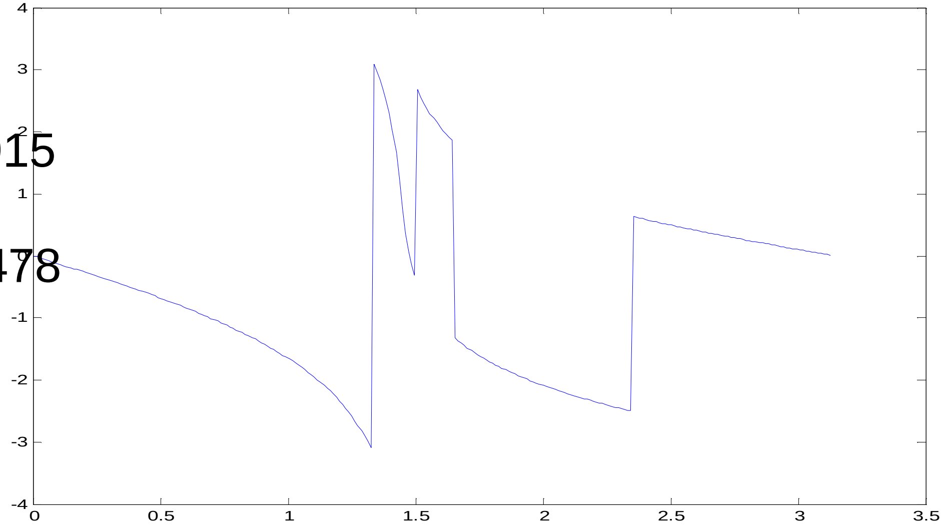


Risultati

N = 6, wn=0.45

b = 0.1403 0.1979 0.4193 0.3915
0.4193 0.1979 0.1403

a = 1.0000 -0.8988 2.0700 -1.1478
1.1006 -0.3087 0.1103



aumenta l'ordine

Filtri FIR - sintesi con MATLAB

Si specifica l'andamento desiderato del modulo della risposta in frequenza per valori discreti di ω:

$|H_d(\omega_i)|$ i=1,2.....Q

Fissato l'ordine del filtro FIR, il suo modulo si può scrivere come

$|H(\omega)|=|b_0+ b_1 e^{-j\omega}+ b_2 e^{-j2\omega}+\ldots. b_Q e^{-j\omega M}|$

Si cercano i valori dei coefficenti $b_{o\ldots.}$ b_Q che minimizzano la distanza tra $H_d(\omega i)$ e $H(\omega i)$, ad es. la somma dei quadrati delle distanze (si usano per queste tecniche di ottimizzazione)

Si richiede fase lineare – quindi $b_k = b_{Q-k}$

Funzione MATLAB FIRPM

FIRPM

Parks-McClellan optimal equiripple FIR filter design

b = firpm(n,f,a) returns row vector b containing the n+1 coefficients of the order n FIR filter whose frequency-amplitude characteristics match those given by vectors f and a.

The output filter coefficients in b obey a symmetry relation

Vectors f and a specify the frequency-magnitude characteristics of the filter:

- f is a vector of pairs of normalized frequency points, specified in the range between 0 and 1, where 1 corresponds to the Nyquist frequency. The frequencies must be in increasing order.
- a is a vector containing the desired amplitudes at the points specified in f.
- The desired amplitude at frequencies between pairs of points ($f(k)$, $f(k+1)$) for k odd is the line segment connecting the points ($f(k)$, $a(k)$) and ($f(k+1)$, $a(k+1)$).
- The desired amplitude at frequencies between pairs of points ($f(k)$, $f(k+1)$) for k even is unspecified. The areas between such points are transition or "don't care" regions.
- f and a must be the same length. The length must be an even number.

b=firpm(n,f,a,w) uses the weights in w to weight the error. W has one entry per band (so it is half the length of f and a) which tells how much emphasis to put on minimizing the error in each band relative to the other bands.

[b,err] = firpm(...) returns the maximum ripple height in err

[b,err,res] = firpm(...) returns a structure res with the following fields.

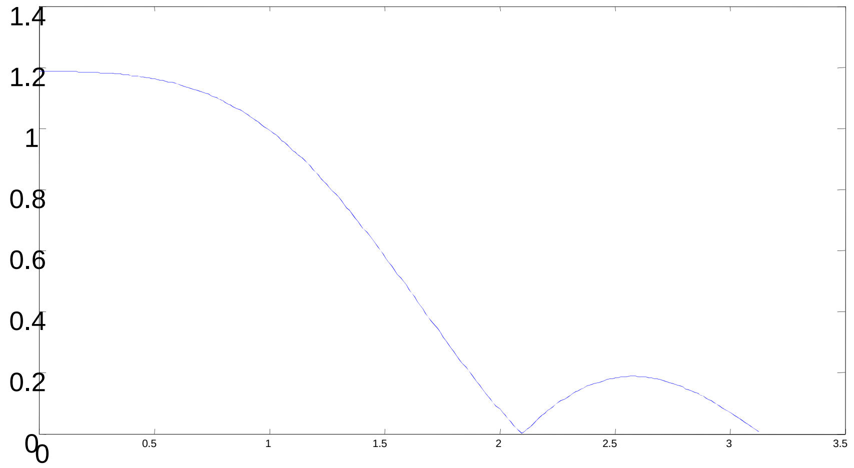
res.fgrid	**Frequency grid vector used for the filter design optimization**
res.des	Desired frequency response for each point in res.fgrid
res.wt	Weighting for each point in opt.fgrid
res.H	Actual frequency response for each point in res.fgrid
res.error	Error at each point in res.fgrid (res.des-res.H)
res.iextr	Vector of indices into res.fgrid for extremal frequencies
res.fextr	Vector of extremal frequencies

Esempio

FC = 8kHz
Filtro passabasso, banda passante 0-1.6kHz
banda oscura 2.4-4kHz

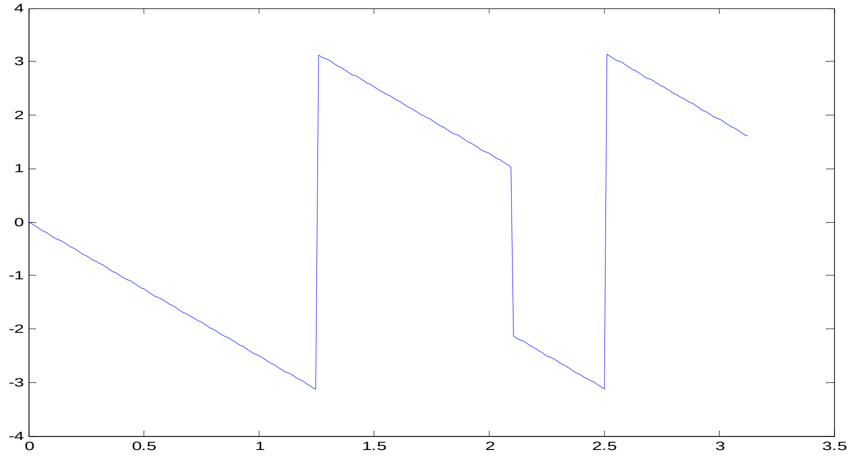
File MATLAB
Wp=1600/4000
Ws=2400/4000
f=[0 wp ws 1]
m=[1 1 0 0]

[b,err]=firpm(5,f,m)



Risultati
b = -0.0807 0.1967 0.4777 0.4777
0.1967 -0.0807

err = 0.1874



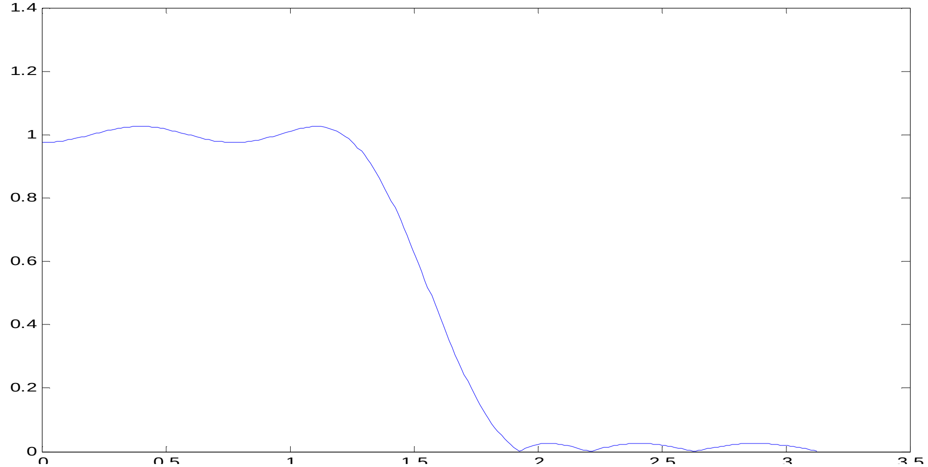
Se aumenta l'ordine del filtro

FC = 8kHz
Filtro passabasso, banda passante 0-1.6kHz
banda oscura 2.4-4kHz

File MATLAB

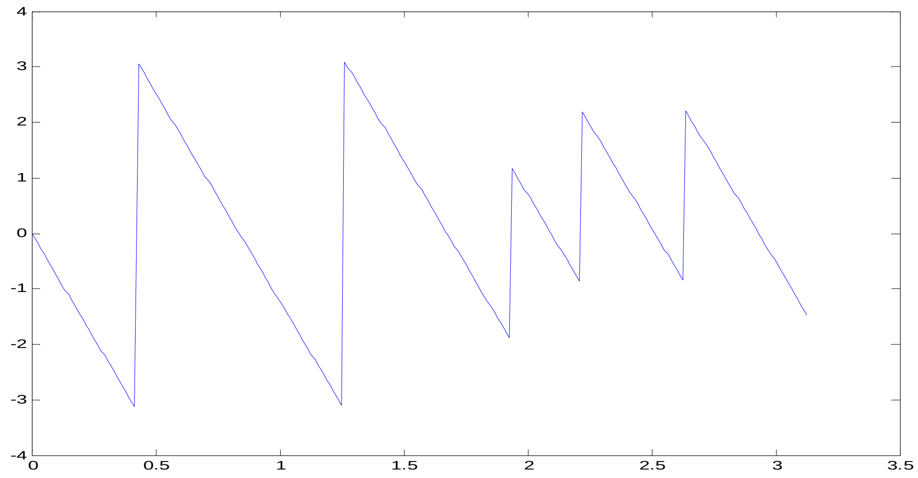
```
Wp=1600/4000
Ws=2400/4000
f=[0 wp ws 1]
m=[1 1 0 0]

[b,err]=firpm(15,f,m)
```



Risultati

```
b = 0.0117   -0.0237    0.0288    0.0356
   -0.0539   -0.0820    0.1460    0.4480
    0.4480    0.1460   -0.0820   -0.0539
    0.0356    0.0288   -0.0237   -0.0117
err = 0.0257
```



diminuisce l'errore

Se si riduce la banda di transizione:

FC = 8kHz

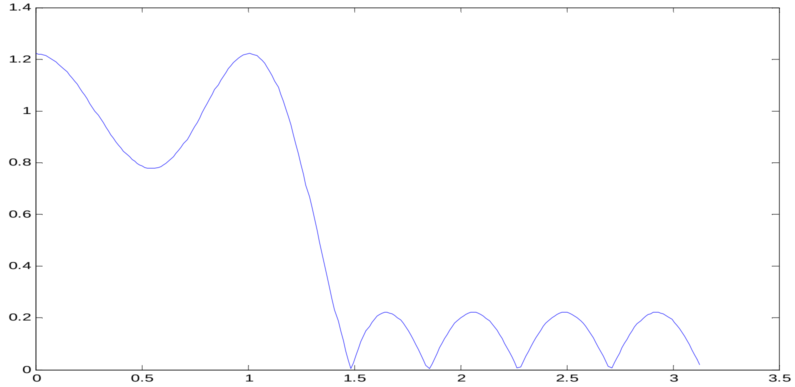
Filtro passabasso   banda passante 0-1.6kHz

banda oscura   1.8-4kHz

File MATLAB

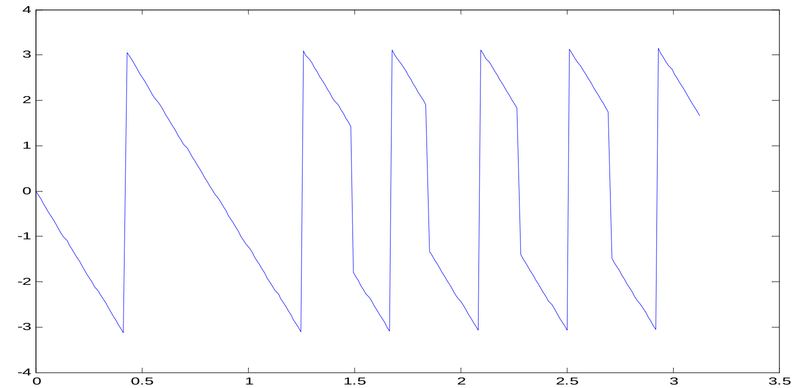
```
Wp=1600/4000
Ws=1800/4000
f=[0 wp ws 1]
m=[1 1 0 0]

[b,err]=firpm(15,f,m)
```



Risultati

```
b =  -0.0542    0.1122    0.0821   -0.0075
     -0.0869   -0.0231   0.1935    0.3946
      0.3946    0.1935   -0.0231   -0.0869
     -0.0075    0.0821    0.1122   -0.0542
err =  0.2214
```



aumenta l'errore

Se si riduce il ripple in banda passante

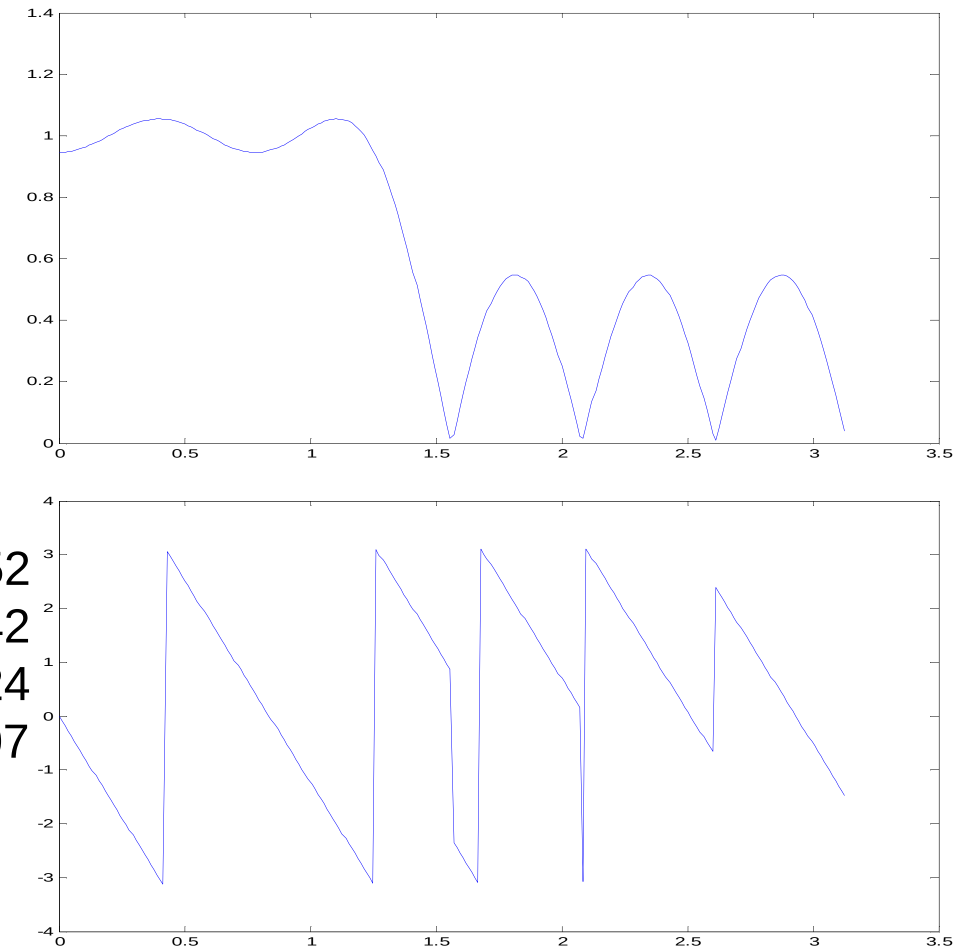
FC = 8kHz

Filtro passabasso banda passante 0-1.6kHz

banda oscura 1.8-4kHz

File MATLAB

```
Wp=1600/4000
Ws=1800/4000
f=[0 wp ws 1]
m=[1 1 0 0]
w=[10   1]
[b,err]=firpm(15,f,m,w)
```



Risultati

b =	-0.0107	-0.1113	0.1829	-0.0452
	-0.1124	-0.0240	0.1991	0.3942
	0.3942	0.1991	-0.0240	-0.1124
	-0.0452	0.1829	-0.1113	-0.0107

err = 0.5466

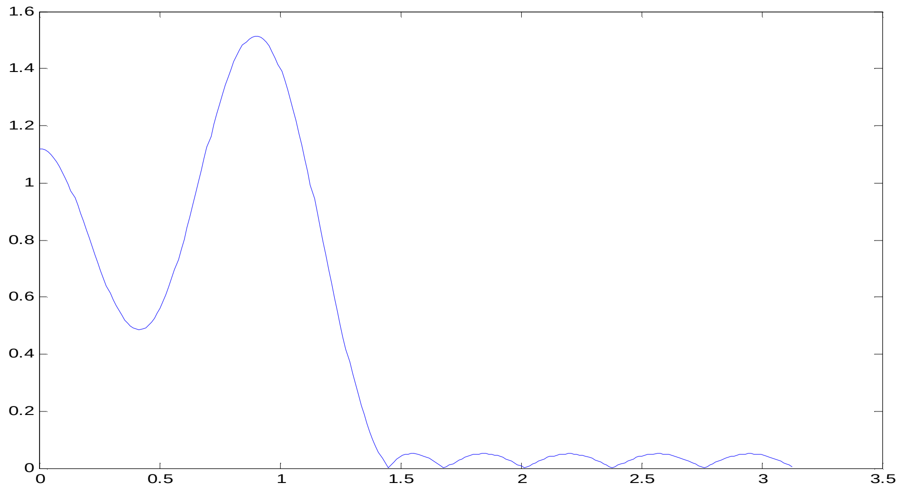
aumenta il ripple in banda oscura

Se si riduce il ripple in banda oscura

FC = 8kHz
Filtro, passabasso banda passante 0-1.6kHz
banda oscura 1.8-4kHz

File MATLAB

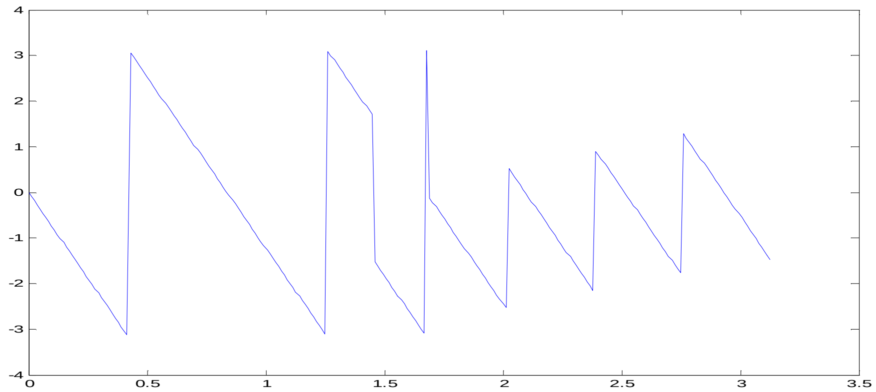
```
Wp=1600/4000
Ws=1800/4000
f=[0 wp ws 1]
m=[1 1 0 0]
w=[1  10]
[b,err]=firpm(15,f,m,w)
```



Risultati

b=0.0786 0.1127 0.0699 -0.0510
-0.1336 -0.0549 0.1697 0.3681
0.3681 0.1697 -0.0549 -0.1336
-0.0510 0.0699 0.1127 0.078

err = 0.5134



aumenta il ripple in banda passante