Corso di Elaborazione numerica dei segnali
Anno accademico 2002/2003.
Prof. Pasquale Daponte.

Esercitazione n°3
Filtri digitali.

1) Filtro digitale con un solo polo.

Un semplice filtro digitale del primo ordine ha la seguente funzione di trasferimento:

$$H\left(z^{-1}\right)=\frac{1-|a|}{1+az^{-1}}$$

Osservare la risposta in frequenza del filtro su indicato al variare del parametro *a* nel range [-1,1].

In MATLAB è possibile costruire un modello di un sistema lineare a partire dalla funzione di trasferimento indicata, nel caso di sistemi tempo-discreto, in termini della zeta-trasformata. Il comando:

```
SYS=filt(num,den)
```

restituisce un sistema lineare che ha la funzione di trasferimento indicata dai parametri numeratore (num) e denominatore (den). E' possibile specificare in tempo di campionamento aggiungendolo come parametro alla funzione:

```
SYS=filt(num,den,Ts)
```

La funzione

```
H=freqresp(SYS,W)
```

calcola la risposta in frequenza del sistema lineare alle frequenze specificate dal vettore W. Tali frequenze devono essere reali ed espresse in rad/s.

Il seguente script consente di calcolare e visualizzare la risposta del filtro proposto per a=0.5:

```
a=-0.5;
P=filt([1-abs(a)],[1 a]); % definizione del sistema
w=linspace(0,pi,500);
H=freqresp(P,w);         % calcolo della risposta in frequenza
subplot(2,1,1);
plot(w,abs(H(:)));
xlabel('Pulsazione discreta [rad]');
ylabel('Ampiezza');
title('Risposta in frequenza');
subplot(2,1,2);
plot(w,angle(H(:)));
xlabel('Pulsazione discreta [rad]');
ylabel('Fase [rad]');
```

In figura 1 è mostrata la risposta in frequenza ottenuta dallo script precedente.

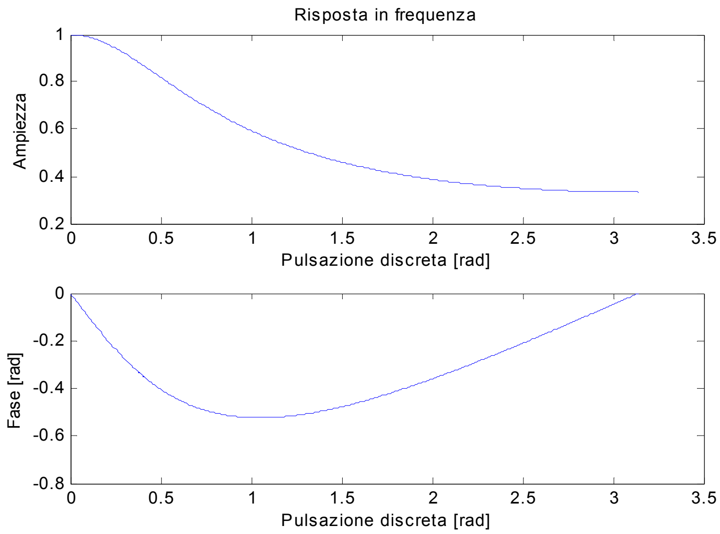

Figura 1 - Risposta in frequenza di un filtro con un solo polo.

2) Filtro digitale con un solo zero

Osservare la risposta in frequenza del filtro con un solo zero:

$$H\left(z^{-1}\right)=\frac{1+az^{-1}}{1+|a|}$$

al variare del parametro a tra -1 e 1.

Lo script dell'esercizio precedente deve essere modificato come segue:

```
a=-0.5;
P=filt([1 a],[1]);          % definizione del sistema
w=linspace(0,pi,500);
H=freqresp(P,w);            % calcolo della risposta in frequenza
subplot(2,1,1);
plot(w,abs(H(:)));
xlabel('Pulsazione discreta [rad]');
ylabel('Ampiezza');
title('Risposta in frequenza');
subplot(2,1,2);
plot(w,angle(H(:)));
xlabel('Pulsazione discreta [rad]');
ylabel('Fase [rad]');
```

La risposta in frequenza corrispondente è mostrata in figura 2.

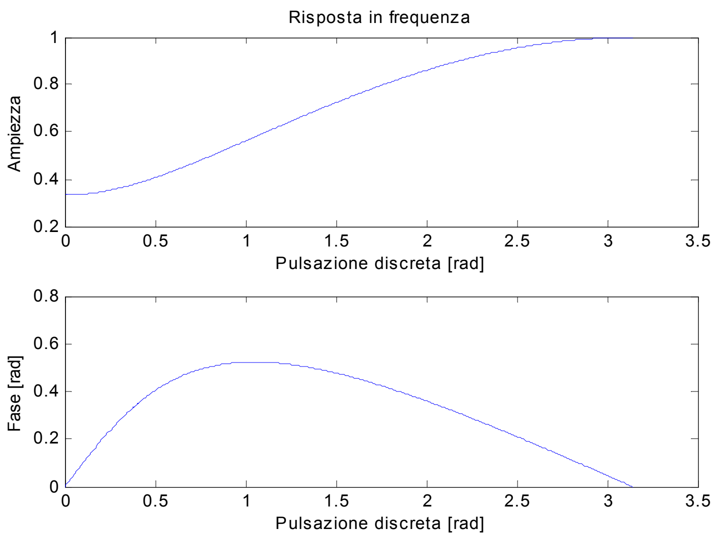

Figura 2 - Risposta in frequenza di un filtro con un solo zero.

3) Filtri FIR a fase lineare.

Visualizzare la fase della risposta in frequenza dei seguenti filtri FIR:

$$H1\left(z^{-1}\right) = 0.5 + 0.7z^{-1} + 0.3z^{-2} + 0.4z^{-3}$$
$$H2\left(z^{-1}\right) = 0.0416 + 0.4584z^{-1} + 0.4584z^{-5} + 0.0416z^{-3}$$

Lo script seguente calcola e visualizza la fase della risposta in frequenza dei filtri H1 e H2.

```
f1=[0.5 0.7 0.3 0.4];
f2=[0.0416 0.4584 0.4584 0.0416];
P1=filt(f1,[1]);             % definizione del sistema
w=linspace(0,pi,500);
H1=freqresp(P1,w);           % calcolo della risposta in
subplot(1,2,1);               % frequenza
plot(w,angle(H1(:)));
xlabel('Pulsazione discreta [rad]');
ylabel('Fase [rad]');
title('Filtro FIR b=[0.5 0.7 0.3 0.4]');
P2=filt(f2,[1]);             % definizione del sistema
H2=freqresp(P2,w);           % calcolo della risposta in
subplot(1,2,2);               % frequenza
plot(w,angle(H2(:)));
xlabel('Pulsazione discreta [rad]');
ylabel('Fase [rad]');
title('Filtro FIR b=[0.0416 0.4584 0.4584 0.0416]');
```

La figura 3 mostra il risultato dello script.

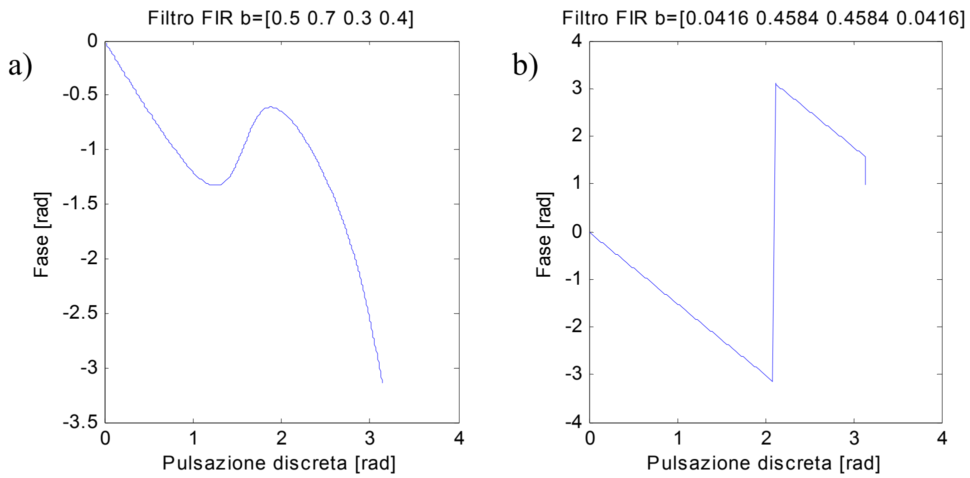

Figura 3 - Fase della risposta in frequenza di un filtro a coefficienti asimmetrici (a) ed uno a coefficienti simmetrici (b).

Il filtro H2 ha fase lineare. Esso presenta infatti una simmetria dei coefficienti.

4) Progetto di filtri FIR mediante allocazione di zeri.

Progettare un filtro FIR a fase lineare[1], mediante l'allocazione di zeri, la cui risposta in frequenza possa essere approssimata dalla seguente espressione.

$$H(\omega)=\begin{cases}0 & 0\leq\omega\leq 0.2\pi\\ 1 & 0.4\pi\leq\omega\leq 0.6\pi\\ 0 & 0.8\pi\leq\omega\leq\pi\end{cases}$$

Si può cominciare allocando zeri nei punti:

cos(0.2π)+sin(0.2π), cos(0.2π)-sin(0.2π), cos(0.8π)+sin(0.8π), cos(0.8π)-sin(0.8π),
cos(0)+sin(0), cos(π)+sin(π)

```
0.8090+0.5878i, 0.8090-0.5878i, -0.8090+0.5878i, -0.8090-0.5878i
1,    -1
```

situati sulla circonferenza di raggio unitario.

Per smussare la risposta in frequenza nella banda passante, si possono inserire degli zeri sull'asse immaginario, per esempio nei punti:

```
-0.5i,    0.5i,          -2i, 2i
```

Il seguente script visualizza la risposta in frequenza (figura 4) del filtro FIR con gli zeri impostati.

```
r=[0.8090+0.5878i  0.8090-0.5878i  -0.8090+0.5878i...
    -0.8090-0.5878i -1 1 -0.5i 0.5i -2i 2i];
b=poly(r);
P=filt(b,[1]);           % definizione del sistema
w=linspace(0,pi,500);
H=freqresp(P,w);         % calcolo della risposta in frequenza
plot(w,abs(H(:)));
xlabel('Pulsazione discreta [rad]');
ylabel('Ampiezza');
title('Risposta in frequenza');
```

[1] Per ottenere un filtro a fase lineare gli zeri che non appartengono alla circonferenza di raggio unitario devono essere, a due a due l'uno il reciproco dell'altro. Se, per esempio fisso uno zero nel punto 0.5, dovrò fissarne anche uno nel punto 2, affinché il filtro risulti a fase lineare. Questa condizione deriva dal fatto che i coefficienti sono simmetrici.

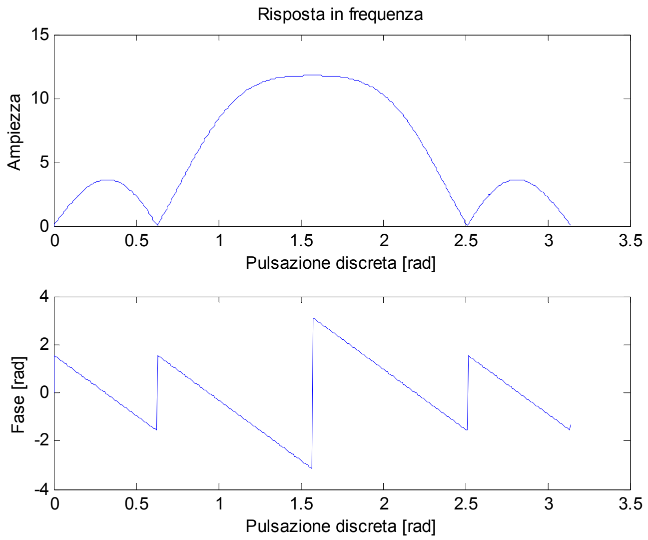

Figura 4 - Risposta in frequenza del filtro progettato.

5) Progetto di filtri FIR passa-basso con il metodo delle finestre.

Un filtro FIR passa-basso può essere progettato finestrando la risposta all'impulso di un filtro ideale. Un filtro passa-basso ideale ha una risposta in frequenza espressa da:

$$H(\omega) = \begin{cases} 1 & 0 \le \omega < \omega_c \\ 0 & \omega_c \le \omega \le \pi \end{cases}$$

Un filtro di questo tipo ha una risposta all'impulso infinita, espressa da:

$$h(n) = \frac{\sin(\omega_c n)}{\pi n}, \qquad n \in]-\infty, \infty[\,,$$

I coefficienti del filtro si possono ottenere finestrando la risposta all'impulso con una finestra rettangolare.

$$h(n) = \frac{\sin\left(\omega_c\left(n - \frac{M-1}{2}\right)\right)}{\pi\left(n - \frac{M-1}{2}\right)} \qquad n \in [0, M\text{-}1]$$

Il seguente script calcola e visualizza la risposta in frequenza ottenuta in questo modo (figura 5).

```
M=20;
n=0:M-1;
h=sin(0.3*pi*(n-(M-1)/2))./(pi*(n-(M-1)/2));     % risposta
P=filt(h,[1]);                                     % all'impulso
w=linspace(0,pi,500);
H=freqresp(P,w);
subplot(2,1,1);
plot(w,abs(H(:)));
xlabel('Pulsazione discreta [rad]');
ylabel('Ampiezza');
title('Risposta in frequenza');
subplot(2,1,2);
plot(w,unwrap(angle(H(:))));
xlabel('Pulsazione discreta [rad]');
ylabel('Fase [rad]');
```

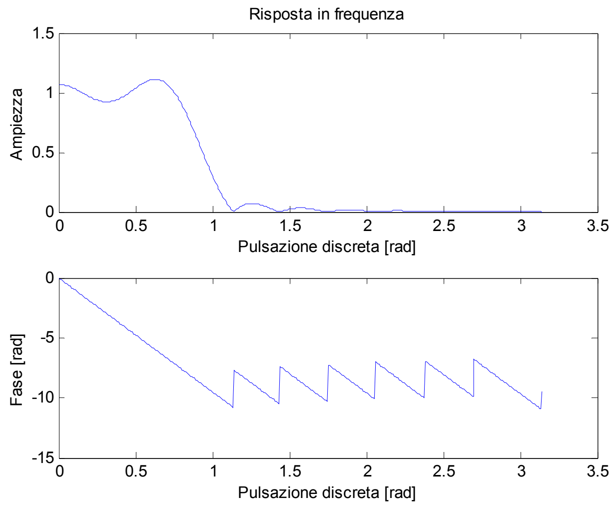

Figura 5 - Risposta in frequenza di un filtro FIR passa-basso progettato con il metodo delle finestre.

Per limitare il fenomeno delle oscillazioni di Gibbs, si può finestrare la risposta all'impulso con una delle finestre utilizzate per limitare la dispersione spettrale, nel calcolo dell'FFT. Se, per esempio si utilizzala finestra di Hanning, lo script si modifica come segue:

```
M=20;
n=0:M-1;
h=sin(0.3*pi*(n-(M-1)/2))./(pi*(n-(M-1)/2));    % risposta
                                                 % all'impulso
wind=0.5-0.5*cos(2*pi*n/M-1);               % finestra di Hanning
hw=h.*wind;
P=filt(hw,[1]);
w=linspace(0,pi,500);
H=freqresp(P,w);
subplot(2,1,1);
plot(w,abs(H(:)));
xlabel('Pulsazione discreta [rad]');
ylabel('Ampiezza');
title('Risposta in frequenza');
subplot(2,1,2);
plot(w,unwrap(angle(H(:))));
xlabel('Pulsazione discreta [rad]');
ylabel('Fase [rad]');
```

In figura 6 è mostrata la risposta in frequenza del filtro considerato ottenuto con la finestra di Hanning.

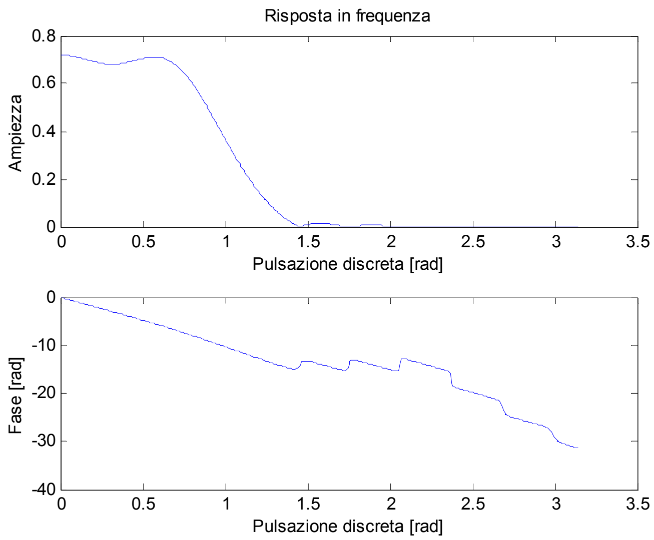

Figura 6 –Filtro FIR progettato con una finestra di Hanning.

6) Filtri IIR.

Calcolare e visualizzare l'ampiezza della risposta in frequenza dei seguenti filtri IIR:

$$H_1(z^{-1}) = \frac{0.0181 + 0.0543z^{-1} + 0.0543z^{-2} + 0.0181z^{-3}}{1 - 1.763z^{-1} + 1.1829z^{-2} - 0.2781z^{-3}} \quad \text{filtro di Butterworth}$$

$$H_2(z^{-1}) = \frac{0.0115 + 0.0344z^{-1} + 0.0344z^{-2} + 0.0115z^{-3}}{1 - 2.1378z^{-1} + 1.7693z^{-2} - 0.5398z^{-3}} \quad \text{filtro di Chebyshev}$$

$$H_3(z^{-1}) = \frac{0.0626 - 0.0059z^{-1} - 0.00593z^{-2} + 0.0626z^{-3}}{1 - 2.1279z^{-1} + 1.7848z^{-2} - 0.5434z^{-3}} \quad \text{filtro Ellittico}$$

Lo script seguente realizza le risposte in frequenza, mostrate in figura 7.

```
b1 =[0.0181    0.0543    0.0543    0.0181];
a1 =[1.0000   -1.7600    1.1829   -0.2781];
b2 =[0.0115    0.0344    0.0344    0.0115];
a2 =[1.0000   -2.1378    1.7693   -0.5398];
b3 =[0.0626   -0.0059   -0.0059    0.0626];
a3 =[1.0000   -2.1279    1.7848   -0.5434];

P1=filt(b1,a1);
P2=filt(b2,a2);
P3=filt(b3,a3);

w=linspace(0,pi,500);

H1=freqresp(P1,w);
subplot(3,1,1);
plot(w,abs(H1(:)));
xlabel('Pulsazione discreta [rad]');
ylabel('Ampiezza');
title('Risposta in frequenza di H1');
axis([0 pi 0 1]);

H2=freqresp(P2,w);
subplot(3,1,2);
plot(w,abs(H2(:)));
xlabel('Pulsazione discreta [rad]');
ylabel('Ampiezza');
title('Risposta in frequenza di H2');
axis([0 pi 0 1]);

H3=freqresp(P3,w);
subplot(3,1,3);
plot(w,abs(H3(:)));
xlabel('Pulsazione discreta [rad]');
ylabel('Ampiezza');
title('Risposta in frequenza di H3');
axis([0 pi 0 1]);
```

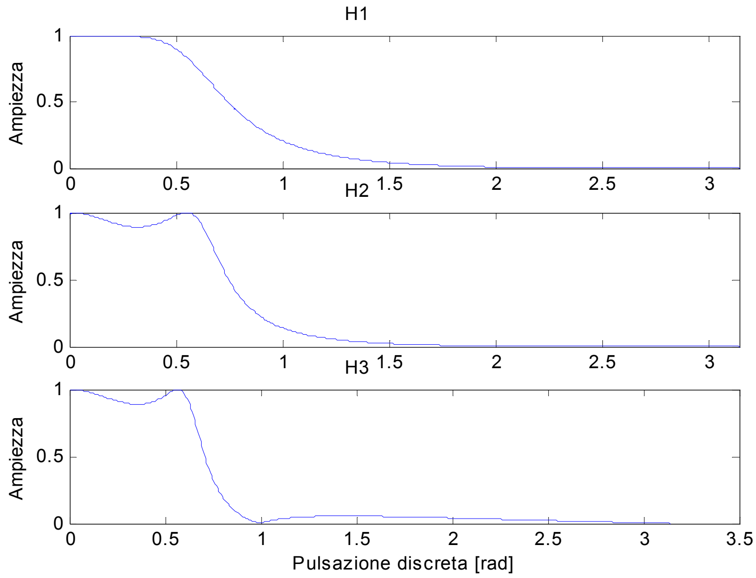

Figura 7 - Risposta in frequenza di filtri IIR

7) Realizzazione di un filtro.

In MATLAB, la funzione

```
y=filter(B,A,x)
```

filtra i dati contenuti nel vettore x, con il filtro descritto dai vettori A (denominatore della funzione di trasferimento) e B (numeratore della funzione di trasferimento) e crea il vettore dei dati filtrati y. Il filtro è eseguito, implementando l'equazione alle differenze:

```
a(1)*y(n) = b(1)*x(n) + b(2)*x(n-1) + ... + b(nb+1)*x(n-nb)
                      - a(2)*y(n-1) - ... - a(na+1)*y(n-na)
```

dove na ed nb sono rispettivamente le dimensioni dei vettori A e B.

Si supponga di avere un segnale composto da due sinusoidi, una a una frequenza f_1 di 1.2 kHz, l'altra a f_2=3 kHz, di ampiezza rispettivamente 2 e 1.5. Si voglia filtrare il segnale per estrarre la sola sinusoide a 1kHz. Progettare un filtro FIR con il metodo delle finestre e filtrare il segnale. Si ponga la frequenza di campionamento f_s pari a 10 kHz e l'ordine del filtro pari a 50.

Si può porre la frequenza wc del filtro nel punto medio tra le frequenze delle due sinusoidi.

$$\omega_c = 2\pi \frac{f_1 + f_2}{f_s} = 2\pi \frac{3000+1200}{20000}$$

Il filtro viene progettato come nell'esercizio precedente, utilizzando una finestra di Hanning.

```
b1 =[0.0181    0.0543    0.0543    0.0181];
a1 =[1.0000   -1.7600    1.1829   -0.2781];

b2 =[0.0115    0.0344    0.0344    0.0115];
a2 =[1.0000   -2.1378    1.7693   -0.5398];

b3 =[0.0626   -0.0059   -0.0059    0.0626];
a3 =[1.0000   -2.1279    1.7848   -0.5434];

P1=filt(b1,a1);
P2=filt(b2,a2);
P3=filt(b3,a3);

w=linspace(0,pi,500);

H1=freqresp(P1,w);
subplot(3,1,1);
plot(w,abs(H1(:)));
ylabel('Ampiezza');
title('H1               ');
axis([0 pi 0 1]);
```

```
H2=freqresp(P2,w);
subplot(3,1,2);
plot(w,abs(H2(:)));
ylabel('Ampiezza');
title('H2                    ');
axis([0 pi 0 1]);

H3=freqresp(P3,w);
subplot(3,1,3);
plot(w,abs(H3(:)));
xlabel('Pulsazione discreta [rad]');
ylabel('Ampiezza');
title('H3                    ');
```

In figura 8 sono mostrati il segnale di ingresso e quello di uscita al filtro.

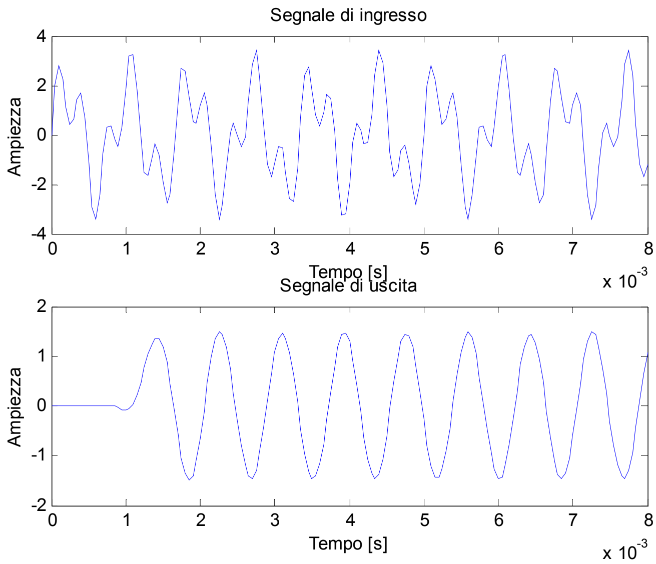

Figura 8 - Segnali di ingresso e di uscita al filtro.