

1 Esercizi al calcolatore proposti

Questa raccolta contiene una serie di esercizi proposti che possono essere agevolmente risolti con l'uso di MATLAB. Lo scopo degli esercizi è duplice: rendere il lettore più familiare con i concetti di base della teoria e mostrargli come MATLAB possa essere convenientemente usato per l'analisi e l'elaborazione dei segnali.

2 Filtri Numerici

In questa sezione, considereremo soprattutto alcuni esercizi relativi al filtraggio dei segnali. L'attenzione verrà posta al filtraggio dei segnali a tempo discreto, e si supporrà dunque di avere a che fare con segnali definiti su $Z(T)$.

La relazione ingresso-uscita di un filtro definito su $Z(T)$ con risposta impulsiva $h(t)$, $t \in Z(T)$, risulta come noto data dall'operazione di convoluzione

$$y(nT) = T \sum_{k=-\infty}^{+\infty} h(kT)x(nT - kT) = T \sum_{k=-\infty}^{+\infty} x(kT)h(nT - kT). \quad (1)$$

Cominceremo con l'introdurre alcuni concetti fondamentali per la realizzazione e la comprensione dei filtri a tempo discreto. Il lettore potrà rivelarne l'analogia con concetti relativi ad altre discipline, in cui si considerano sistemi a tempo continuo.

2.0.1 Stabilità BIBO di un filtro numerico

Un filtro numerico si dice *stabile* in senso BIBO (Bounded Input-Bounded Output) se l'uscita ad un ingresso limitato è anch'essa limitata. Si ricorda che il segnale $x(nT)$ è limitato se esiste un numero finito M tale che

$$|x(t)| \leq M, \quad \text{per ogni } t \in Z(T). \quad (2)$$

Vengono ora analizzate le condizioni per cui la trasformazione lineare (1) risulta stabile in senso BIBO. Come è logico attendersi, le condizioni riguardano la risposta impulsiva $h(nT)$. Si ha infatti

$$\begin{aligned} |y(nT)| &\leq \sum_{k=-\infty}^{+\infty} T|h(kT)||x(nT - kT)| \\ &\leq M \sum_{k=-\infty}^{+\infty} T|h(kT)|, \end{aligned}$$

dove si è sfruttato il fatto che, per un ingresso limitato, vale la (2). Se dunque la somma dei moduli della risposta impulsiva del filtro è finita, così risulta l'uscita $y(nT)$. La relazione precedente stabilisce una condizione sufficiente per la stabilità BIBO. Tale condizione è anche necessaria: se infatti fosse

$$\sum_{k=-\infty}^{+\infty} T|h(kT)| = +\infty \quad (3)$$

e $h(kT)$ non fosse limitata, basterebbe prendere l'ingresso $x(nT) = \delta_{Z(T)}(nT)$ per ottenere un'uscita $y(nT) = h(nT)$ non limitata. Se infine valesse la (3) e $h(kT)$ fosse limitata, basterebbe prendere l'ingresso $x(nT-kT) = h^*(kT)/|h(kT)|$, per $|h(kT)| \neq 0$ e 0 altrove, per avere $y(nT) = +\infty$ all'istante nT . Possiamo dunque concludere con il seguente Teorema.

Teorema 1 *Un filtro sui tempi discreti caratterizzato dalla risposta impulsiva $h(t), t \in Z(T)$ è stabile in senso BIBO se e solo se*

$$\sum_{k=-\infty}^{+\infty} T|h(kT)| < +\infty.$$

Dunque, un filtro risulta stabile in senso BIBO se la sua risposta impulsiva decade in maniera sufficientemente rapida, per nT che tende a più o meno infinito, in modo che valga la

$$\sum_{k=-\infty}^{+\infty} T|h(kT)| < +\infty$$

Si introduce la seguente nomenclatura: un filtro a tempo discreto si dice di tipo FIR (Finite Impulse Response) se la sua risposta impulsiva è di durata *finita*. Viceversa, esso si dice di tipo IIR (Infinite Impulse Response) se la sua risposta impulsiva ha durata non limitata.

2.0.2 Risposta in frequenza razionale

Una classe molto importante di filtri a tempo discreto è la classe dei filtri causali a risposta in frequenza razionale, tali cioè che la trasformata di Fourier $H(f), f \in R/Z(T)$, della risposta impulsiva $h(t), t \in Z(T)$, assume la forma

$$H(f) = \frac{T \sum_{k=0}^r b_k e^{-j2\pi f k T}}{\sum_{k=0}^q a_k e^{-j2\pi f k T}}, \quad (4)$$

dove si può porre, senza perdita di generalità, $a_0 = 1$. Nel caso infatti fosse $a_0 \neq 1$, basterebbe dividere numeratore e denominatore per a_0 . La risposta in frequenza $H(f)$ risulta, come è noto e come si vede immediatamente dalla (4), una funzione periodica in f di periodo $F_s = 1/T$.

Il termine *razionale* risulta più comprensibile se, anziché considerare la trasformata di Fourier (4), ci riferiamo alla trasformata zeta della risposta impulsiva, detta anche *funzione di trasferimento del filtro*,

$$H_z(z) \triangleq T \sum_{n=-\infty}^{+\infty} h(nT)z^{-n},$$

la quale risulta un rapporto fra polinomi in z^{-1} . Infatti, ricordando che la trasformata di Fourier si ottiene dalla trasformata zeta ponendo $z = \exp(j2\pi fT)$, nell'ipotesi di esistenza di entrambe, dalla (4) si ottiene

$$H_z(z) = \frac{T \sum_{k=0}^r b_k z^{-k}}{\sum_{k=0}^q a_k z^{-k}}. \quad (5)$$

Vedremo come la classe dei filtri a risposta in frequenza razionale possa essere realizzata mediante algoritmi numerici estremamente semplici.

Consideriamo dapprima alcuni semplici esempi.

Esempio 1. La sequenza esponenziale.

Consideriamo il filtro a tempo discreto con risposta impulsiva *causale* $h(nT) = p^n 1_0(nT)$. Il filtro è stabile in senso BIBO se risulta $|p| < 1$. In tali ipotesi, la risposta in frequenza è una funzione razionale e risulta

$$H(f) = \frac{T}{1 - pe^{-j2\pi fT}}, \quad f \in R/Z(1/T).$$

La trasformata zeta vale

$$H(z) = \frac{T}{1 - pz^{-1}} \quad (6)$$

e presenta un polo in $z = p$ all'interno del cerchio unitario del piano complesso e uno zero nell'origine. La regione di convergenza della trasformata zeta è la regione $|z| > |p|$.

Consideriamo invece un filtro con risposta in frequenza

$$H(f) = \frac{T}{1 - pe^{-j2\pi fT}}, \quad f \in R/Z(1/T),$$

dove però sia $|p| > 1$. In questo caso, l'antitrasformata risulta, come è immediato verificare, un segnale *anti-causale* ed uguale a

$$h(nT) = -p^n 1_0(-nT - T).$$

Si tratta anche in questo caso di un filtro stabile in senso BIBO. La trasformata zeta vale

$$H(z) = \frac{T}{1 - pz^{-1}}$$

ma questa volta la regione di convergenza risulta $|z| < |p|$.

Infine, un filtro con risposta impulsiva causale $h(nT) = p^n 1_0(nT)$, con $|p| \geq 1$, non è ovviamente stabile in senso BIBO. La sua trasformata zeta assume ancora la forma (6) e converge per $|z| > |p|$, mentre la sua trasformata di Fourier non esiste (almeno in senso ordinario).

Un facile esercizio, risolvibile applicando più volte la regola di derivazione della trasformata di Fourier, permetterebbe di stabilire che la sequenza causale

$$h(nT) = \frac{(n+k-1)!}{n!(k-1)!} p^n 1_0(nT), \quad nT \in Z(T), \quad (7)$$

stabile in senso BIBO per $|p| < 1$, ha trasformata di Fourier

$$H(f) = \frac{T}{(1 - pe^{-j2\pi fT})^k}, \quad f \in R/Z(T) \quad (8)$$

e trasformata zeta, convergente per $|z| > |p|$,

$$H_z(z) = \frac{T}{(1 - pz^{-1})^k}. \quad (9)$$

In conclusione, le (7)-(9), valide per $k \geq 1$, forniscono il legame segnale-trasformata relativo a sequenze *causali* e *stabili* con andamento esponenziale nel dominio del tempo e forma razionale nel dominio della trasformata. In particolare, dato che ci limitiamo a considerare sequenze causali, la trasformata zeta (9) ha un polo, di molteplicità k , all'interno del cerchio unitario e uno zero di molteplicità k nell'origine. \square

Consideriamo ora il caso della generica trasformata zeta razionale (5). Il calcolo della anti-trasformata è banale se $q = 0$. In tale caso, infatti, $H_z(z)$ si riduce ad un polinomio in z^{-1} e i coefficienti della risposta impulsiva risultano uguali a $h(nT) = b_n$ per $n = 0, \dots, r$, e 0 altrove. La risposta impulsiva è dunque, in questo caso, di tipo FIR.

Nel caso $q \neq 0$, il modo di procedere è quello di sviluppare la (5) in *frazioni parziali*: possiamo scrivere

$$\begin{aligned} H_z(z) &= T \frac{b_0 + b_1 z^{-1} + \dots + b_r z^{-r}}{1 + a_1 z^{-1} + \dots + a_m z^{-q}} \\ &= T \frac{b_0 + b_1 z^{-1} + \dots + b_r z^{-r}}{(1 - p_1 z^{-1})^{m_1} \dots (1 - p_s z^{-1})^{m_s}}, \end{aligned} \quad (10)$$

dove p_i è il polo i -esimo della trasformata zeta, e m_i la rispettiva molteplicità, con $m_1 + \dots + m_s = q$. Supponendo $r < q$ nella espressione della trasformata zeta¹, la (10) può essere scritta nella forma

$$H_z(z) = \sum_{j=1}^{m_1} \frac{A_{1j}}{(1 - p_1 z^{-1})^j} + \dots + \sum_{j=1}^{m_s} \frac{A_{sj}}{(1 - p_s z^{-1})^j}, \quad (11)$$

dove, come è noto,

$$A_{ij} = \frac{1}{(m_i - j)!} \frac{1}{(-p_i)^{m_i - j}} \frac{d^{m_i - j}}{d(z^{-1})^{m_i - j}} \left[H_z(z) (1 - p_i z^{-1})^{m_i} \right]_{z=p_i}. \quad (12)$$

¹Se così non fosse, basterebbe prendere il resto della divisione del numeratore per il denominatore, scomponendo $H_z(z)$ nella somma di un polinomio in z^{-1} e di una funzione razionale della forma voluta.

Si noti che la derivata viene fatta rispetto alla variabile $w = z^{-1}$. Si noti inoltre che nel caso di un polo p_i con molteplicità unitaria, il coefficiente A_{i1} si calcola semplicemente dalla (12) valutando la funzione $H_z(z)(1 - p_i z^{-1})$ per $z = p_i$.

Possiamo calcolare l'antitrasformata di $H_z(z)$ usando i risultati dell'esempio precedente. Se ne conclude che, *se tutti i poli p_i sono in modulo minori di 1*, l'antitrasformata della $H_z(z)$ corrisponde alla combinazione lineare di sequenze con andamento esponenziale della forma (7). Nel caso fosse $r \geq q$, alla combinazione di esponenziali si aggiungerebbe nella risposta impulsiva un contributo di durata finita, corrispondente all'antitrasformata di un polinomio in z^{-1} .

Esempio 2.

Si consideri un filtro causale con funzione di trasferimento

$$H_z(z) = T \frac{1.5 + 0.5z^{-1} + 0.2z^{-2}}{1 - 0.7z^{-1} + 0.1z^{-2}}. \quad (13)$$

Si ricava facilmente che i poli di $H_z(z)$ valgono $z = 0.5$ e $z = 0.2$ e sono quindi interni al cerchio di raggio unitario. Se ne deduce che $H_z(z)$ è la funzione di trasferimento di un filtro causale e stabile. Applicando la scomposizione in frazioni parziali, dopo avere calcolato il resto della divisione fra numeratore e denominatore, si trova

$$H_z(z) = T \left[2 + \frac{5.5}{1 - 0.5z^{-1}} - \frac{6}{1 - 0.2z^{-1}} \right].$$

La risposta impulsiva del filtro risulta pertanto di tipo IIR

$$h(nT) = 2T\delta_{Z(T)}(nT) + 5.5 \cdot 0.5^n 1_0(nT) - 6 \cdot 0.2^n 1_0(nT).$$

□

3 Equazioni alle differenze

Si consideri un filtro con una risposta in frequenza razionale, come nell'equazione (4). La relazione ingresso-uscita del filtro, espressa nel dominio della frequenza, risulta

$$Y(f) = H(f)X(f), \quad f \in R/Z(T), \quad (14)$$

dove $Y(f)$ e $X(f)$ sono la trasformata di Fourier dell'uscita e dell'ingresso, rispettivamente. Moltiplicando entrambi i membri della (14) per il denominatore di $H(f)$ e ricordando la regola di traslazione per la trasformata di Fourier, si ottiene nel dominio del tempo la seguente relazione ricorsiva fra $y(nT)$ e $x(nT)$

$$y(nT) = - \sum_{k=1}^q a_k y((n-k)T) + T \sum_{k=0}^r b_k x((n-k)T). \quad (15)$$

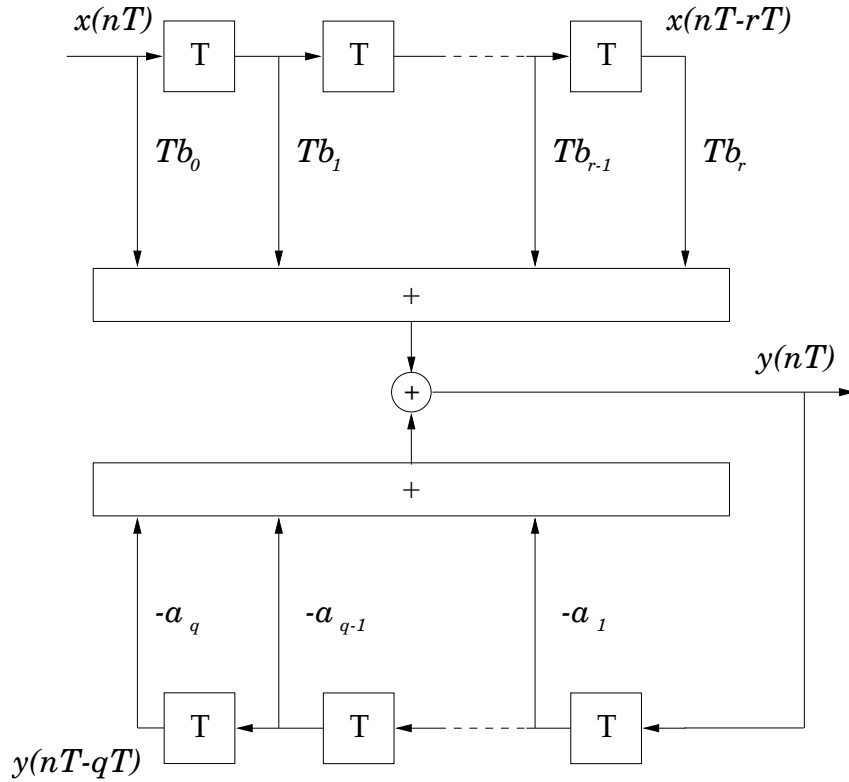


Figure 1: Schema per la realizzazione dell'equazione a differenze finite.

L'equazione *lineare alle differenze finite* (15), corrispondente ad una funzione di trasferimento razionale, può facilmente essere realizzata mediante un algoritmo numerico, come esemplificato nello schema di Fig. 1.

La sua struttura suggerisce anche la forma normalizzata

$$y(nT) = - \sum_{k=1}^q a_k y((n-k)T) + \sum_{k=0}^r b'_k x((n-k)T),$$

in cui vengono definiti i coefficienti $b'_k = Tb_k$.

Esempio 3. La funzione di trasferimento (13) corrisponde alla seguente equazione alle differenze

$$y(nT) = 0.7y((n-1)T) - 0.1y((n-2)T) + 1.5Tx(nT) + 0.5Tx((n-1)T) + 0.2Tx((n-2)T). \quad (16)$$

Come si vede, l'uscita al tempo nT dipende dai valori precedenti dell'ingresso e dell'uscita. Se però l'ingresso $x(nT)$ è nullo per $nT < n_0T$, anche l'uscita $y(nT)$ risulta nulla per $nT < n_0T$, dato che il filtro

è causale e risulta $h(nT) = 0$ per $nT < 0$. In tali ipotesi, il sistema (16) evolve a partire da *condizioni iniziali* nulle a partire dall'istante nT_0 .

È possibile scrivere il seguente programma MATLAB per il calcolo dell'uscita da $n_0T = 0$ a $nT = 100T$ corrispondente all'ingresso $x(nT) = \sin(2\pi f_0 nT)1_0(nT)$, con $T = 1$, $f_0 = 0.1T$.

```
% condizioni iniziali -caricamento dello ‘‘stato’’ del filtro
```

```
xs=[0,0];
ys=[0,0];
r1=length(xs);
q1=length(ys);
```

```
% istante iniziale
n0=0;
```

```
n=n0;
T=1;
```

```
while (n>=n0 & n<=100),
```

```
% nuovo ingresso
```

```
  xnT=sin(2*pi*0.1*n);
```

```
% calcolo dell'uscita: l'indice n+1 e' relativo al campione y(nT)
```

```
  y(n+1)=[0.7 -0.1]*ys' + [1.5 0.5 0.2]*[xnT xs]';
```

```
% aggiornamento dello stato
```

```
  ys=[y(n+1),   ys(1:q1-1)];
  xs=[xnT,      xs(1:r1-1)];
```

```
  n=n+1;
end;
```

□

Nell'esempio precedente abbiamo visto come possiamo utilizzare la (15) per il calcolo dell'uscita di un filtro causale con funzione di trasferimento razionale quando l'ingresso è nullo prima di un certo istante. Come possiamo calcolare l'uscita nel caso in cui l'ingresso sia diverso da zero, in generale, da meno infinito a più infinito? Ovviamente, saremo interessati dal punto di vista operativo a conoscere l'uscita a partire da un certo istante t di osservazione. Per poter utilizzare la (15) dovremmo però conoscere le condizioni iniziali, specificate anche dai valori precedenti dell'uscita, che purtroppo in generale non conosciamo dato che stiamo cercando di calcolarla.

La soluzione del problema si ottiene in maniera semplice se il segnale di ingresso $x(t)$, $t \in Z(T)$, ha una durata convenzionale *praticamente* limitata. In questo caso, infatti, è sufficiente pensare che il segnale sia nullo al di fuori di un certo intervallo ed usare la (15) per calcolare l'uscita del filtro con la precisione voluta.

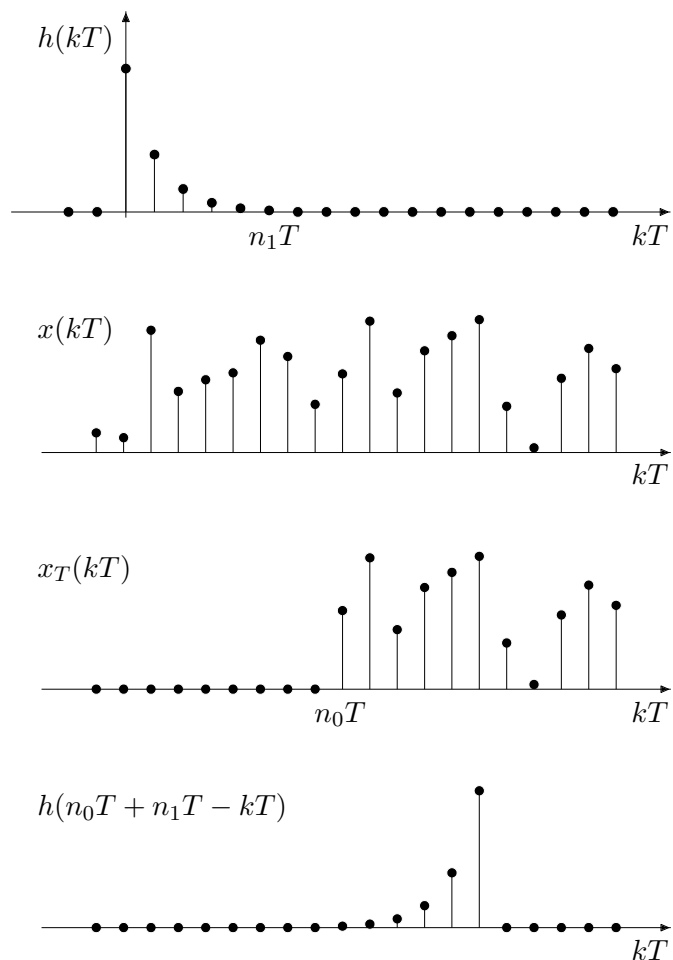


Figure 2: Nel calcolo della convoluzione, all'istante $n_0T + n_1T$ le uscite corrispondenti a $x(t)$ e $x_T(t)$ coincidono (praticamente).

Nel caso invece il segnale abbia un andamento persistente, come ad esempio avviene per i segnali sinusoidali, il segnale a gradino, o anche, in generale, per le realizzazioni dei processi aleatori, è utile fare la seguente osservazione. Come abbiamo visto, la risposta impulsiva $h(nT)$ di un filtro causale stabile con risposta in frequenza razionale ha un andamento dato dalla somma

di una sequenza di durata finita e di un certo numero di sequenze esponenziali. Nei limiti di una precisione prefissata, tale risposta impulsiva può considerarsi in ogni caso trascurabile a partire da un certo istante $t = n_1T$ che dipende dalla velocità di decadimento degli esponenziali, e quindi dal modulo dei poli della funzione di trasferimento. Più i poli sono piccoli in modulo, più velocemente si esaurisce la risposta impulsiva. Di conseguenza, se approssimiamo l'ingresso $x(nT)$ con una sua versione troncata $x_T(nT) = x(nT)1_0(nT - n_0T)$, le uscite $y(nT) = x * h(nT)$ e $y_T(nT) = x_T * h(nT)$ sono praticamente uguali per $t \geq n_0T + n_1T$ (vedi Fig. 2). Il calcolo di $y_T(nT)$ può dunque essere effettuato utilizzando l'equazione alle differenze (15) a partire da condizioni iniziali nulle: dopo un numero di campioni di uscita pari a n_1T , ovvero dopo l'esaurimento del *transitorio* del filtro, i campioni di $y_T(nT)$ coincidono praticamente con quelli che si sarebbero ottenuti filtrando l'ingresso originario. Si dice in questo caso che, dopo l'esaurimento del transitorio, l'uscita del filtro è in *regime permanente*. Si noti l'analogia con la nomenclatura ed i concetti usati, anche in altre discipline, per i segnali a tempo continuo.

3.1 L'uso di MATLAB per il progetto dei filtri

Abbiamo visto nella sezione precedente come sia possibile realizzare tramite un algoritmo numerico un filtro a tempo discreto con risposta in frequenza razionale. Esistono in realtà dei criteri di organizzazione dei calcoli che rendono la realizzazione del filtro più robusta, ad esempio qualora sia necessario utilizzare processori che utilizzano un'aritmetica a precisione finita. Il programma MATLAB fornisce con il *Signal Processing Toolbox* un insieme di procedure per il filtraggio a tempo discreto. In particolare, la funzione `filter(b,a,x)`, supponendo uguale ad 1 il primo elemento `a(1)` del vettore `a`, filtra i dati contenuti nel vettore `x` di ingresso realizzando l'equazione alle differenze

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

Si noti che i coefficienti del vettore `b(i)` devono essere posti uguali a Tb_{i-1} , secondo quanto richiesto nella (15)².

La funzione `[H,f]=freqz(b,a,N,Fs)` calcola N campioni complessi equispaziati fra 0 e $F_s/2$, con $F_s = 1/T$, della trasformata di Fourier corrispondente alla funzione di trasferimento razionale

$$H_z(z) = \frac{b(1) + b(2)z^{-1} + \dots + b(nb+1)z^{-nb}}{1 + a(2)z^{-1} + \dots + a(na+1)z^{-na}}.$$

In particolare, gli N valori della risposta in frequenza vengono posti nel vettore `H`, mentre i rispettivi valori della frequenza, in Hz, vengono posti nel vettore `f`. Si noti che la trasformata

²Gli algoritmi di MATLAB assumono che $T=1$. Ci si può facilmente ricondurre al caso generale supponendo di riferirci ai coefficienti normalizzati Tb_k .

di Fourier $H(f) = H_z(e^{j2\pi fT})$ ha simmetria hermitiana, e quindi ci si può limitare a specificarla tra 0 e $F_s/2$.

Per l'analisi di una funzione di trasferimento, sono utili anche le procedure **residue** e **roots** che calcolano, rispettivamente, la scomposizione in frazioni parziali di un rapporto di polinomi e le radici di un polinomio.

Un argomento di fondamentale importanza, cui qui accenneremo soltanto parzialmente, riguarda il *progetto* di filtri numerici. Il problema che qui ci si pone è il seguente: assegnata una determinata risposta in frequenza del filtro, come è possibile determinare i coefficienti b_i e a_i di una risposta in frequenza razionale che approssimi l'andamento voluto? Esistono a questo proposito molte tecniche di progetto, relative sia a filtri FIR ($q = 0$ nella (5)), che a filtri IIR.

Tipicamente, i filtri che si desidera realizzare hanno un andamento della risposta in frequenza che approssima uno degli andamenti ideali di Fig. 3, relativi a filtri *passa-basso*, *passa-alto*, *passa-banda* e *elimina-banda*, rispettivamente. Nella Fig. 3, le frequenze sono normalizzate rispetto alla frequenza di campionamento F_s . Trattandosi di filtri reali, è sufficiente rappresentare l'andamento della risposta in frequenza fra $f = 0$ e $f = F_s/2$. Come vedremo, i vari metodi di progetto si limitano molto spesso a specificare solamente il *modulo* della risposta in frequenza desiderata.

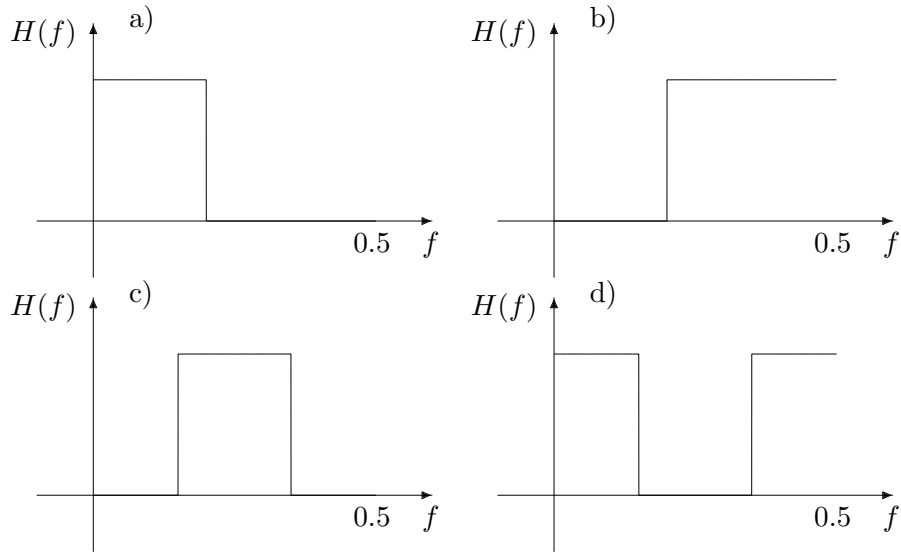


Figure 3: Andamento dei filtri ideali a) passa-basso, b) passa-alto, c) passa-banda e d) elimina-banda

Trattandosi di un problema di approssimazione, occorrerà specificare nel progetto le tolleranze ammesse nella realizzazione del filtro. A questo proposito, si consideri la Fig. 4. In

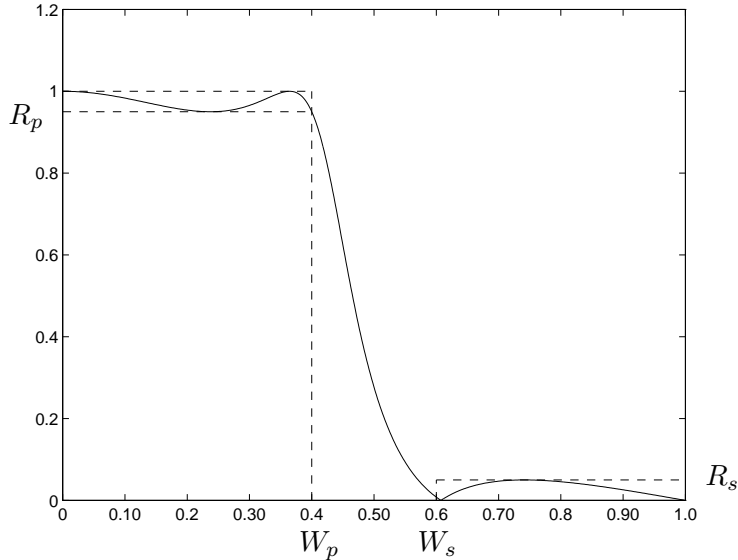


Figure 4: Parametri di progetto di un filtro passa-basso.

essa, viene rappresentato il modulo della risposta in frequenza (l'asse delle ascisse è parametrizzato rispetto a f/F_s) di un tipico filtro numerico che approssima un filtro passa-basso ideale. I parametri da considerare sono:

1. la *banda passante*, specificata dalla frequenza W_p . Si richiede che il filtro abbia modulo della risposta in frequenza circa uguale ad 1 per $0 \leq f \leq W_p$. Per un filtro passabanda, andrà in genere specificato un intervallo di frequenze entro il quale si desidera che il filtro abbia modulo della trasformata di Fourier unitario;
2. la *banda oscura*, specificata in questo caso dalla frequenza W_s . Si richiede che il filtro abbia modulo della risposta in frequenza circa uguale a 0 per $W_p \leq f \leq F_s/2$. L'intervallo $[W_p, W_s]$ specifica la cosiddetta *banda di transizione* del filtro;
3. il *ripple* in banda passante, quantificato tramite il valore R_p (spesso espresso in dB) che permette di determinare l'errore rispetto all'andamento desiderato;
4. il *ripple* in banda attenuata, specificato tramite il valore massimo R_s del modulo della risposta in frequenza del filtro in banda attenuata.

Progetto di filtri IIR in MATLAB. Il programma MATLAB fornisce varie procedure per il progetto di filtri IIR ($q > 0$ nella (5)). Considereremo in particolare il caso dei filtri *ellittici*, i quali forniscono una approssimazione uniforme della risposta in frequenza sia in banda passante che in banda oscura. Si può dimostrare che i filtri ellittici sono ottimi, nel senso che, fissati i valori W_p , R_p e R_s , la banda di transizione $W_s - W_p$ è la minima possibile per un dato grado q del denominatore del filtro (q viene detto *l'ordine* del filtro).

Il progetto dei filtri ellittici passa-basso in MATLAB viene effettuato utilizzando le due procedure `ellipord` e `ellip` del Signal Processing Toolbox. La sintassi di `ellipord` è la seguente:

```
[N, Wn] = ellipord(Wp, Ws, RpdB, RsdB).
```

In ingresso alla funzione vanno specificate le frequenze W_p e W_s (vedi Fig. 4), normalizzate rispetto a $F_s/2$. Inoltre, occorre specificare i valori di R_p e R_s , in dB. L'uscita della funzione fornisce il valore N dell'ordine q del filtro che soddisfa le specifiche, e il valore della frequenza naturale Wn da usare per l'effettivo progetto del filtro, effettuato con la procedura

```
[b,a] = ellip(N, RpdB, RsdB, Wn).
```

Nei vettori \mathbf{b} ed \mathbf{a} , si trovano i coefficienti del numeratore e denominatore della risposta in frequenza razionale (5) che soddisfa le specifiche. Si noti che i valori nel vettore \mathbf{b} includono la moltiplicazione per il quanto temporale T . Il metodo di progetto con filtri ellittici non permette di specificare la caratteristica di *fase* del filtro.

Esempio 4. Supponiamo che la frequenza di campionamento del sistema numerico sia $F_s = 8$ kHz, e si voglia progettare un filtro passa-basso con banda passante da 0 a 1.6 kHz e banda oscura da 2.4 kHz a 4 kHz= $F_s/2$. Si supponga di imporre $R_p = 0.95$ e $R_s = 0.05$. Il progetto e la visualizzazione della risposta in frequenza del filtro, mostrata nelle figure 5 e 6, viene effettuata con il seguente programma MATLAB.

```
% Parametri di progetto

Fs=8000;
RpdB=-20*log10(0.95);
RsdB=-20*log10(0.05);
Wp=1600/Fs*2;
Ws=2400/Fs*2;

% Progetto del filtro

[N,Wn]=ellipord(Wp, Ws, RpdB, RsdB);
[b,a]=ellip(N, RpdB, RsdB, Wn);

% Visualizzazione del filtro

[H,f]=freqz(b,a,512,Fs);
```

```
plot(f,abs(H));
plot(f,angle(H));
```

Si noti dalla Fig. 5 che il filtro soddisfa effettivamente le specifiche. Nella Fig. 6, si noti la discontinuità di π attorno alla frequenza 2.5 kHz, dovuta al cambiamento di segno di $H(f)$ (corrispondente al punto angoloso nell'andamento del modulo). La fase è rappresentata modulo 2π . L'ordine del filtro è risultato $q = 3$, e quindi i vettori \mathbf{b} e \mathbf{a} hanno dimensione 4. Il filtro può essere utilizzato per l'elaborazione dell'ingresso posto nel vettore \mathbf{x} mediante il comando `filter(b,a,x)`. \square

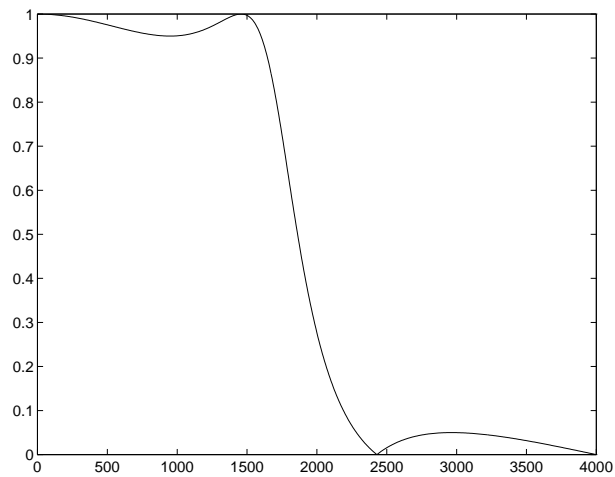


Figure 5: Modulo della risposta in frequenza del filtro ellittico (f in Hz).

Le funzioni `ellipord` e `ellip` possono essere utilizzate per il progetto di filtri passa-alto, passa-banda e elimina-banda. Si considerino in particolare i seguenti esempi esplicativi.

Esempio 5. Progetto di un filtro passa-alto ($F_s = 8$ kHz) con banda oscura $[0,1.5]$ kHz e banda passante $[2.5,4]$ kHz, $R_p = 0.99$ e $R_s = 0.01$.

```
% Parametri di progetto
```

```
Fs=8000;
RpdB=-20*log10(0.99);
RsdB=-20*log10(0.01);
Wp=2500/Fs*2;
Ws=1500/Fs*2;
```

```
% Progetto del filtro
```

```
[N,Wn]=ellipord(Wp, Ws, RpdB, RsdB);
```

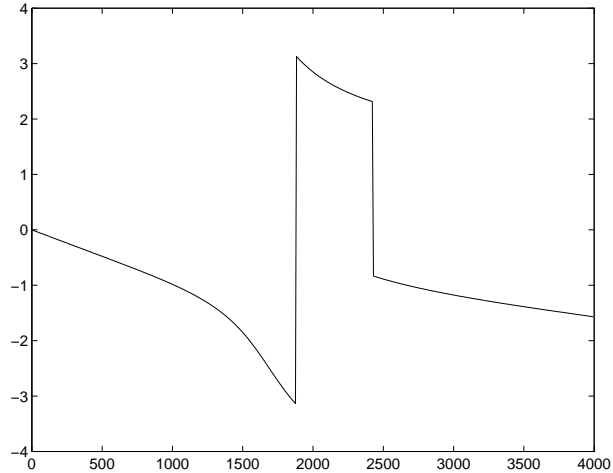


Figure 6: Fase (radianti) della risposta in frequenza del filtro ellittico (f in Hz)

```
[b,a]=ellip(N, RpdB, RsdB, Wn, 'high');
```

Si noti che $W_p > W_s$ e l'uso del parametro 'high' nella chiamata a `ellip`. L'ordine del filtro è risultato $q = 4$. □

Esempio 6. Progetto di un filtro passa-banda ($F_s = 8$ kHz) con banda oscura negli intervalli $[0,1]$ kHz e $[3,4]$ kHz e banda passante $[1.5,2.5]$ kHz, $R_p = 0.95$ e $R_s = 0.01$.

```
% Parametri di progetto
```

```
Fs=8000;
RpdB=-20*log10(0.95);
RsdB=-20*log10(0.01);
Wp=[1500/Fs*2, 2500/Fs*2];
Ws=[1000/Fs*2, 3000/Fs*2];
```

```
% Progetto del filtro
```

```
[N,Wn]=ellipord(Wp, Ws, RpdB, RsdB);
[b,a]=ellip(N, RpdB, RsdB, Wn);
```

Si noti che W_p e W_s sono dei vettori, con componenti le frequenze che delimitano la banda passante e le bande oscure. L'ordine del filtro risulta pari a due volte il valore N ritornato da `ellipord`. In questo caso, è risultato $N=4$ e $q = 8$. □

Esempio 7. Progetto di un filtro stop-banda ($F_s = 8$ kHz) con banda oscura [1.5,2.5] kHz e banda passante negli intervalli [0,1] kHz e [3,4] kHz, $R_s = 0.95$ e $R_p = 0.01$.

```
% Parametri di progetto

Fs=8000;
RpdB=-20*log10(0.95);
RsdB=-20*log10(0.01);
Ws=[1000/Fs*2, 3000/Fs*2];
Wp=[1500/Fs*2, 2500/Fs*2];

% Progetto del filtro

[N,Wn]=ellipord(Wp, Ws, RpdB, RsdB);
[b,a]=ellip(N, RpdB, RsdB, Wn, 'stop');
```

Si noti che W_p e W_s sono dei vettori, con componenti le frequenze che delimitano le bande passanti e la banda oscura. Si noti inoltre l'uso del parametro 'stop' in ingresso alla procedura `ellip`. L'ordine del filtro risulta anche in questo caso pari a due volte il valore N ritornato da `ellipord`. È risultato in questo esempio $N=4$ e $q = 8$. □

Altre procedure per il progetto di filtri IIR disponibili in MATLAB sono `butter`, `cheby1`, `cheby2`, `besself`, `invfreqz`. In particolare, `invfreqz` permette il progetto di filtri di cui si specifica sia il modulo che la fase.

Progetto di filtri FIR in MATLAB. I filtri FIR hanno alcuni svantaggi rispetto ai filtri IIR ma anche alcuni importanti vantaggi. In generale, l'ordine di un filtro FIR (ovvero il parametro r nella (5)) risulta in generale molto più grande dell'ordine di un filtro IIR che soddisfa le stesse specifiche. D'altro canto, la durata del transitorio di un filtro FIR è finita e pari alla durata della risposta impulsiva del filtro. Inoltre, esistono metodi di progetto efficienti che garantiscono che il filtro FIR abbia esattamente *fase lineare*. Come è noto, questo garantisce che una sinusoide in ingresso al filtro subisce in uscita un ritardo indipendente dalla frequenza. Se dunque la risposta in frequenza del filtro ha l'espressione

$$H(f) = e^{-j2\pi ft_0} H_0(f),$$

dove $H_0(f)$ vale 1 nella banda passante, e 0 altrove³ e il segnale di ingresso ha un'estensione spettrale contenuta nella banda passante, l'uscita ha trasformata di Fourier

$$Y(f) = H(f)X(f) = e^{-j2\pi ft_0} X(f).$$

³Dovendo $H(f)$ essere periodica di periodo $1/T$, questo può avvenire solamente se t_0 è un multiplo di T .

Nel dominio del tempo, risulta $y(t) = x(t - t_0)$ e l'uscita è una versione non distorta (secondo Heaviside) del segnale di ingresso.

Un filtro FIR causale di durata NT a fase lineare (a tratti) soddisfa la condizione

$$h(nT) = h((N - 1 - n)T), \quad n = 0, \dots, (N - 1)T.$$

La risposta impulsiva è dunque *simmetrica*, come evidenziato nelle Fig. 7a e 7b nel caso di N pari e N dispari, rispettivamente. In questo caso, la risposta in frequenza del filtro

$$H(f) = T \sum_{n=0}^{N-1} h(nT) e^{-j2\pi f n T}$$

risulta, come è immediato verificare

$$H(f) = \begin{cases} T e^{-j2\pi f T \frac{N-1}{2}} \left[h\left(\frac{NT-T}{2}\right) + 2 \sum_{n=0}^{\frac{N-3}{2}} h(nT) \cos\left(2\pi f T \left(n - \frac{N-1}{2}\right)\right) \right], & N \text{ dispari,} \\ T e^{-j2\pi f T \frac{N-1}{2}} \left[2 \sum_{n=0}^{\frac{N-2}{2}} h(nT) \cos\left(2\pi f T \left(n - \frac{N-1}{2}\right)\right) \right], & N \text{ pari.} \end{cases}$$

La risposta in frequenza è dunque uguale al prodotto di una funzione *reale* per un fattore a fase lineare. Eventuali discontinuità di π nella fase possono presentarsi in corrispondenza di cambi di segno della funzione reale (da cui la dizione *fase lineare a tratti*). Per N dispari, è possibile progettare i coefficienti $h(nT)$ in modo che la funzione reale sia circa uguale ad 1 in un certo intervallo di frequenze e 0 altrove, cosicché un ingresso con estensione spettrale contenuta nella banda passante si ritrova ritardato in uscita di $(N - 1)/2$ campioni e praticamente non distorto.

Se N è pari, $(N - 1)/2$ non è un numero intero e l'uscita non può essere in nessun caso una semplice versione ritardata dell'ingresso a tempo discreto. Tuttavia, se l'ingresso è ottenuto per campionamento su $Z(T)$ di una forma d'onda continua $x_a(t), t \in R$, nell'ipotesi in cui l'estensione spettrale del segnale campionato sia contenuta nella banda passante del filtro, l'uscita risulta pari a $y(nT) = x_a(nT - T(N - 1)/2)$. Essa si ottiene dunque per campionamento della versione del segnale a tempo continuo ritardato di $t_0 = T(N - 1)/2$.

Nota. Se N è pari risulta in ogni caso $H(F_s/2) = 0, F_s = 1/T$. Infatti, $H(F_s/2)$ risulta uguale all'area del segnale

$$h(nT) e^{-j2\pi \frac{F_s}{2} n T} = (-1)^n h(nT)$$

che risulta nulla, data la simmetria di $h(nT)$ (vedi Fig. 7a).

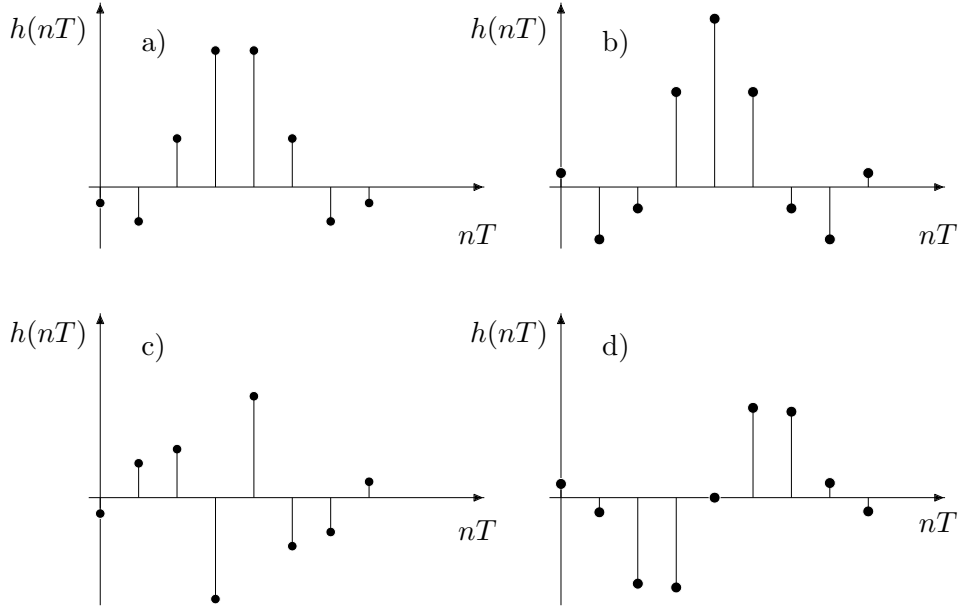


Figure 7: Andamenti della risposta impulsiva di filtri FIR simmetrici e antisimmetrici.

Non è dunque conveniente cercare di progettare un filtro FIR passa-alto a fase lineare con un numero pari di campioni.

Un'altra classe di filtri FIR la cui risposta in frequenza presenta un fattore a fase lineare è quella dei filtri antisimmetrici, la cui risposta impulsiva verifica la condizione

$$h(nT) = -h((N - 1 - n)T).$$

Gli andamenti tipici sono mostrati nelle Fig. 7c e 7d per N pari e N dispari rispettivamente.

La risposta in frequenza risulta

$$H(f) = \begin{cases} -jT e^{-j2\pi fT \frac{N-1}{2}} \left[2 \sum_{n=0}^{\frac{N-3}{2}} h(nT) \sin \left(2\pi fT \left(n - \frac{N-1}{2} \right) \right) \right], & N \text{ dispari,} \\ -jT e^{-j2\pi fT \frac{N-1}{2}} \left[2 \sum_{n=0}^{\frac{N-2}{2}} h(nT) \sin \left(2\pi fT \left(n - \frac{N-1}{2} \right) \right) \right], & N \text{ pari.} \end{cases}$$

La risposta in frequenza risulta dunque il prodotto di un fattore a fase lineare e di una funzione puramente *immaginaria*. In particolare, il fattore $-j$ contribuisce ad una fase costante di $-\pi/2$.

Nota. Data l'antisimmetria, l'area della risposta impulsiva è nulla e risulta dunque $H(0) = 0$. Non è dunque conveniente progettare un filtro passa-basso utilizzando filtri FIR antisimmetrici.

Per N dispari, risulta anche $H(F_s/2) = 0$, per cui il progetto di filtri passa-alto non è consigliabile in questo caso.

Il progetto di filtri FIR simmetrici in MATLAB si effettua utilizzando la procedura

$$\mathbf{b} = \text{remez}(\mathbf{r}, \mathbf{f}, \mathbf{M}, \mathbf{w}).$$

Il filtro fornito dall'algoritmo di Remez, particolarizzato da Parks e McClellan al progetto di filtri FIR, è ottimo nel senso che minimizza il massimo errore di approssimazione nelle varie bande a parità di specifiche. La risposta in frequenza presenta una oscillazione uniforme nelle varie bande (ed è dunque, come si dice, di tipo *equiripple*). I parametri di ingresso alla procedura sono

1. l'ordine \mathbf{r} del filtro, la cui risposta impulsiva risulta pertanto di lunghezza $N = \mathbf{r} + 1$ campioni;
2. un vettore di frequenze \mathbf{f} , normalizzate rispetto a $F_2/2$, $F_s = 1/T$, che specifica le frequenze che delimitano le bande attenuate e oscure della risposta in frequenza;
3. un vettore \mathbf{M} che specifica le ampiezze desiderate alle frequenze specificate nel vettore \mathbf{f} ;
4. un vettore di pesi \mathbf{w} che specifica il rapporto fra le ampiezze degli errori di approssimazione nelle varie bande. Se il filtro non è soddisfacente in termini di errore *assoluto*, occorre aumentare l'ordine del filtro.

Il vettore \mathbf{b} di $\mathbf{r}+1$ elementi contiene i coefficienti del filtro, che includono la moltiplicazione per il quanto temporale T . Il filtro può essere utilizzato per l'elaborazione dell'ingresso posto nel vettore \mathbf{x} mediante il comando `filter(b,1,x)`.

Esempio 8. Supponiamo che la frequenza di campionamento del sistema numerico sia $F_s = 8$ kHz, e si voglia progettare un filtro FIR simmetrico passa-basso di ordine $r = 18$ con banda passante da 0 a 1.6 kHz e banda oscura da 2.4 kHz a 4 kHz= $F_s/2$. Detto δ_p il massimo errore nel modulo della risposta in frequenza in banda passante e δ_s il massimo errore in banda oscura, si desidera che $\delta_p/\delta_s = 2$, ovvero si tollera un errore doppio in banda passante rispetto alla banda oscura.

Il progetto e la visualizzazione del filtro possono essere ottenuti con i seguenti comandi MATLAB.

```

% Specifiche di progetto.

Fs=8000;
f=[0, 1600, 2400 4000]*2/Fs;
M=[1, 1, 0, 0];
w=[1, 2];
r=18;

% Progetto del filtro

b=remez(r,f,M,w);

% Visualizzazione del filtro

[H,f]=freqz(b,1,512,Fs);
figure;
plot(f,abs(H));
figure;
plot(f,angle(H));

```

Il modulo e la fase della risposta in frequenza del filtro ottenuto sono mostrati nelle Fig. 8 e 9. Si noti che la fase è rappresentata modulo 2π e che si hanno discontinuità di π in banda attenuata, corrispondenti a cambi di segno della risposta in frequenza. Una stima approssimata della lunghezza del filtro FIR con banda di transizione $\nu = W_s - W_p$ (espressa in termini di frequenze normalizzate rispetto a F_s) e con massimi errori δ_p e δ_s in banda passante e banda oscura, può essere ottenuta dalla relazione empirica [4],[5]

$$N \simeq \frac{-10 \log_{10}(\delta_p \delta_s) - 15}{14\nu} + 1.$$

Si può altrimenti usare la procedura MATLAB `remezord`.

□

Esempio 9. Progetto di un filtro passa-alto con $r + 1 = 19$ coefficienti ($F_s = 8$ kHz) con banda oscura $[0,1.5]$ kHz e banda passante $[2.5,4]$ kHz, con un errore massimo in banda passante uguale a quello in banda oscura.

```

% Specifiche di progetto.

Fs=8000;
f=[0, 1500, 2500, 4000]*2/Fs;
M=[0, 0, 1, 1];
w=[1, 1];
r=18;

% Progetto del filtro

```

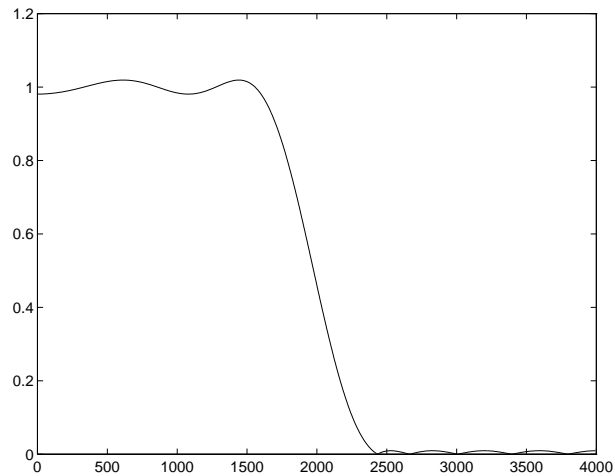


Figure 8: Modulo della risposta in frequenza del filtro FIR simmetrico (f in Hz).

```
b=remez(r,f,M,w);
```

□

Esempio 10. Progetto di un filtro multi banda con $r + 1 = 52$ ($F_s = 8$ kHz) con bande passanti [0,1] kHz, [2,3] kHz e bande oscure [1.2,1.8] kHz, [3.2,4] kHz. Nella banda [0,1] kHz si tollera un errore pari a 1/2 di quello nelle bande [2,3] kHz e [3.2,4] kHz. Nella banda [1.2,1.8] kHz si tollera un errore pari a 1/3 di quello nelle bande [2,3] kHz e [3.2,4] kHz.

```
% Specifiche di progetto.
```

```
Fs=8000;
f=[0, 1000, 1200, 1800, 2000, 3000, 3200, 4000]*2/Fs;
M=[1, 1, 0, 0, 1, 1, 0, 0];
w=[2, 3, 1, 1];
r=51;
```

```
% Progetto del filtro
```

```
b=remez(r,f,M,w);
```

□

Il progetto di filtri antisimmetrici in MATLAB si effettua ancora con la procedura `remez`, usando il parametro `'Hilbert'`. Si considerino i seguenti esempi.

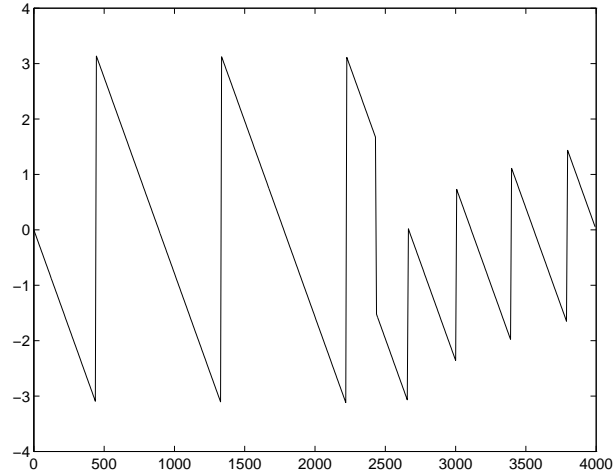


Figure 9: Fase (radianti) della risposta in frequenza del filtro FIR simmetrico (f in Hz).

Esempio 11. Progetto di un filtro passa-alto anti-simmetrico con $r + 1 = 18$ coefficienti ($F_s = 8$ kHz) con banda oscura $[0,1.5]$ kHz e banda passante $[2.5,4]$ kHz, con un errore massimo in banda passante uguale a quello in banda oscura. È opportuno scegliere una lunghezza pari del filtro, trattandosi di un filtro passa-alto antisimmetrico.

% Specifiche di progetto.

```
Fs=8000;
f=[0, 1500, 2500, 4000]*2/Fs;
M=[0, 0, 1, 1];
w=[1, 1];
r=17;
```

% Progetto del filtro

```
b=remez(r,f,M,w,'Hilbert');
```

□

Esempio 12. L'uso della parola 'Hilbert' deriva dal fatto che la fase costante $-\pi/2$ dovuta alla presenza del fattore $-j$ nella risposta in frequenza rende i filtri antisimmetrici adatti al progetto di filtri di Hilbert discreti, con risposta in frequenza di modulo unitario a tutte le frequenze e fase $-\pi/2$ per $0 < f < F_s/2$. Per il progetto di un filtro di Hilbert di ordine 30 si possono utilizzare i seguenti comandi

```
Fs=8000;
```

```
r=30;
```

```
b=remez(r,[.1 0.9],[1 1], 'Hilbert')
```

Il filtro che si ottiene, a parte l'introduzione di un ritardo di 15 campioni, realizza approssimativamente la risposta in frequenza desiderata, come si può vedere con i seguenti comandi

```
N=r+1;
```

```
[H,f]=freqz(b,1,512,Fs);
```

```
figure;
```

```
plot(f,abs(H));
```

```
% Disegna la fase, compensando il fattore corrispondente al ritardo  
% dell'uscita.
```

```
figure;
```

```
plot(f,angle(H.*exp(j*2*pi*f/Fs*(N-1)/2)));
```

□

Esercizi

1. Progettare un filtro passa-basso IIR ellittico che soddisfa le seguenti specifiche ($F_s=2$ Hz): $R_p = 0.98$, $R_s = 0.05$, $W_p = 0.6$ Hz, $W_s = 0.8$ Hz. Si scriva una procedura per il calcolo dell'uscita e la si testi con gli ingressi $x(t) = \delta_{Z(T)}(t)$, $t \in Z(T)$ e $x(t) = \cos(2\pi 0.4t)1_0(t)$, $t \in Z(T)$. Si confrontino i propri risultati con quelli ottenuti usando la procedura `filter`.
2. Progettare un filtro passa-basso FIR a fase lineare che soddisfa le seguenti specifiche ($F_s=2$ Hz): $R_p = 0.98$, $R_s = 0.05$, $W_p = 0.6$ Hz, $W_s = 0.8$ Hz. Si provino vari ordini del filtro finché le specifiche non sono soddisfatte esattamente. Si scriva una procedura per il calcolo dell'uscita e la si testi con gli ingressi $x(t) = \delta_{Z(T)}(t)$, $t \in Z(T)$ e $x(t) = \cos(2\pi 0.4t)1_0(t)$, $t \in Z(T)$. Si confrontino i propri risultati con quelli ottenuti usando la procedura `filter`.
3. Progettare un filtro passa-basso IIR ellittico che soddisfa le seguenti specifiche ($F_s = 1/T = 8$ kHz): $R_p = 0.95$, $R_s = 0.02$, $W_p = 1.84$ kHz, $W_s = 2.48$ kHz. Dopo avere calcolato i poli della funzione di trasferimento con la procedura `roots`, si stimi la durata convenzionale della risposta impulsiva all'1%. Si consideri l'ingresso $x(nT) = 2 \sin(2\pi f_0 nT) + \cos(2\pi f_1 nT)$, con $f_0 = 1$ kHz e $f_1 = 3.2$ kHz e si valuti la risposta a regime permanente del filtro. Si confronti l'uscita con il segnale $x_1(nT) = 2 \sin(2\pi f_0 nT)$.
4. *Importanza della fase lineare.* Progettare un filtro passa-basso IIR ellittico che soddisfa le seguenti specifiche ($F_s = 1/T = 8$ kHz): $R_p = 0.95$, $R_s = 0.02$, $W_p = 2.4$ kHz, $W_s = 2.8$ kHz. Si consideri l'ingresso $x(nT) = 2 \sin(2\pi f_0 nT) + \cos(2\pi f_1 nT)$, con $f_0 = 2.35$ kHz e

$f_1 = 1.5$ kHz e si valuti la risposta a regime permanente del filtro. Si progetti un filtro FIR simmetrico che soddisfa alle stesse specifiche. Dopo avere confrontato l'ordine dei due filtri, si consideri l'uscita in regime permanente all'ingresso $x(nT)$. Si confrontino fra loro le uscite ottenute dai due filtri e l'ingresso $x(nT)$.

5. *Notch filter*. Si consideri un filtro con funzione di trasferimento

$$H_z(z) = C \frac{1 - 2 \cos(2\pi f_0 T) z^{-1} + z^{-2}}{1 - 2\rho \cos(2\pi f_0 T) z^{-1} + \rho^2 z^{-2}}, \quad C = \frac{1 + 2\rho \cos(2\pi f_0 T) + \rho^2}{2 + 2 \cos(2\pi f_0 T)}.$$

Si valutino in forma chiusa i poli e gli zeri della funzione di trasferimento. e si disegni il modulo e la fase della risposta in frequenza al variare di $0 < \rho < 1$ e $0 < f_0 < F_s/2$ con $F_s = 1/T$. In particolare, si valuti la frequenza in cui $H(f)$ si annulla, e si considerino valori di ρ prossimi a 1. Sia $F_s = 8$ kHz. Si progetti un filtro che elimini un eventuale tono sinusoidale a 50 Hz sovrapposto al segnale utile. Tale disturbo potrebbe essere generato da interferenze con la tensione di rete.

6. Nel file

`/home/ts/ESERCIZI/frase_8192.mat`

sono contenuti i campioni di segnale vocale relativi alla frase “*E soprattutto, forse, ciò che gli occhi da soli non avrebbero potuto vedere.*” La frequenza di campionamento è $F_s = 8192$ Hz. Il file può essere caricato con il comando `load` ed i campioni vengono posti nel vettore `orig`.

Il segnale vocale è costituito da una successione di suoni elementari detti *fonemi*: ad esempio alla parola “fonemi” corrisponde la successione dei suoni elementari /f/ /o/ /n/ /e/ /m/ /i/. Inoltre le vocali (e in generale anche le consonanti *vocalizzate*, tipo “l”, “m”) hanno tipicamente un andamento quasi periodico. Questo contrasta con l'andamento irregolare e *rumoroso* delle *fricative*, tipo la “s” o la “f”. Infine, le *plosive*, tipo la “p” o la “b”, sono precedute da una pausa prima dell'emissione del suono. Nella frase che stiamo esaminando, si possono osservare le caratteristiche dei vari segmenti del segnale con il comando `plot`. Se l'elaboratore è fornito di scheda audio, è possibile da MATLAB ascoltare la frase o una sua parte con il comando `sound`.

Nel file

`/home/ts/ESERCIZI/frase_r330.mat`

è contenuta una versione della frase corrotta “in fase di registrazione” aggiungendo al segnale un fastidioso tono a 330 Hz. Progettare un filtro notch che elimini tale disturbo.

7. Con riferimento all'esercizio precedente, nel file

`/home/ts/ESERCIZI/frase_r.mat`

è contenuta una versione della frase corrotta in fase di registrazione con un tono di frequenza non nota. Usando una tecnica di stima spettrale (ad esempio, il periodogramma), si stimi la frequenza della sinusoide sovrapposta al segnale e la si elimini con un filtro notch. *Nota: il filtro notch è molto selettivo, ed è quindi importante determinare accuratamente la frequenza della sinusoide per poterla cancellare efficacemente.*

8. Per ragioni tecniche, si vuole convertire la frequenza di campionamento del segnale vocale dell'esercizio 6 da 8192 Hz alla frequenza doppia 16384 Hz. Si realizzi il convertitore di frequenza mediante interpolazione seguita da opportuno filtraggio.

9. Il segnale telefonico viene usualmente campionato a $F_s = 8$ kHz, ma la sua estensione spettrale può essere limitata senza perdita di intelligibilità a $[-3.33, 3.33]$ kHz. Si progetti uno schema di interpolazione/campionamento per passare dalla frequenza $F_s = 8$ kHz alla frequenza $F'_s = 6.66$ kHz. Occorre dunque interpolare il segnale di un fattore 5 e sottocampionare di un fattore 6 (usando un prefiltro). Il file

`/home/ts/ESERCIZI/frase_8000.mat`

contiene la frase dell'esercizio 6 acquisita con $F_s = 8$ KHz.

Bibliografia

- [1] A. Oppenheim, R. Schaffer, "Digital Signal Processing".
- [2] A. Peled, B. Liu, "Digital Signal Processing".
- [3] G.C. Temes, S.K. Mitra, "Modern Filter Theory and Design".
- [4] R. Crochiere, L. Rabiner, "Multirate Digital Signal Processing".
- [5] L. Rabiner et al., "Some Comparisons between FIR and IIR Digital Filters", Bell Systems Technical Journal, vol. 53, Feb. 1974.
- [6] M. Bellanger, "Digital Processing of Signals".

4 Esercizi sulla FFT

In questa sezione vengono proposti alcuni semplici esercizi riguardanti l'uso della FFT per il calcolo della trasformata di Fourier di segnali a tempo discreto e a tempo continuo.

Si ricorda che se $x(t), t \in Z(T)/Z(NT)$ è un segnale a tempo discreto di periodo NT , la sua trasformata di Fourier ha l'espressione

$$X(kF) = T \sum_{n=0}^{N-1} x(nT) e^{-j2\pi \frac{nk}{N}}, \quad F = \frac{1}{NT}. \quad (17)$$

La trasformata di Fourier $X(kF)$ risulta essere definita su $Z(F)/Z(NF)$ ed è quindi completamente specificata dagli N valori assunti sulla cella elementare $f \in \{0, \dots, (N-1)F\}$. Il segnale $x(nT)$ viene recuperato da $X(kF)$ mediante la formula di inversione

$$x(nT) = F \sum_{k=0}^{N-1} X(kF) e^{j2\pi \frac{nk}{N}}. \quad (18)$$

L'algoritmo di FFT permette di calcolare, a partire dai valori $\{x(0), \dots, x((N-1)T)\}$, la sommatoria (17) in modo efficiente. In particolare, se N è una potenza di 2, la (17) viene calcolata in $\mathcal{O}(N \log_2 N)$ operazioni, il che consente una riduzione drastica rispetto al calcolo diretto che richiede $\mathcal{O}(N^2)$ operazioni.

La procedura MATLAB `X=fft(x,N)` calcola mediante un algoritmo di FFT la sommatoria

$$\mathbf{X}(k+1) = \sum_{n=0}^{N-1} \mathbf{x}(n+1) e^{-j2\pi \frac{nk}{N}}$$

a partire da un vettore di ingresso \mathbf{x} . Il risultato viene posto nel vettore di uscita \mathbf{X} , di dimensione N , che non è vincolato a essere una potenza di 2. Se il vettore di ingresso ha dimensione minore di N , esso viene esteso aggiungendo in coda valori nulli. Si noti che l'algoritmo di MATLAB suppone $T=1$, e che occorre in generale premoltiplicare l'ingresso per T , ponendo $\mathbf{x}(n+1) = Tx(nT)$.

Per l'antitrasformata, si può usare la procedura MATLAB `x=ifft(X)`, che calcola in modo efficiente la sommatoria

$$\mathbf{x}(k+1) = \frac{1}{N} \sum_{n=0}^{N-1} \mathbf{X}(n+1) e^{j2\pi \frac{nk}{N}}$$

a partire dal vettore \mathbf{X} di dimensione N . Nel caso $T \neq 1$, occorre porre $\mathbf{X}(n+1) = X(nT)/T$.

L'algoritmo di FFT può dunque essere utilizzato per il calcolo della trasformata di Fourier di segnali a tempo discreto e periodici. Vedremo di seguito la possibilità di utilizzarla per il calcolo della trasformata di segnali a tempo discreto non periodici e di segnali continui.

Segnali a tempo discreto. Se $x(nT), t \in Z(T)$ è un segnale a tempo discreto, in generale non periodico, con estensione contenuta in $\{0, \dots, (N-1)T\}$, possiamo usare la FFT per calcolare i *campioni* della sua trasformata di Fourier. Infatti, dalla definizione di trasformata, si ottiene

$$X(f) = T \sum_{n=-\infty}^{+\infty} x(nT)e^{-j2\pi fnT} = T \sum_{n=0}^{N-1} x(nT)e^{-j2\pi fnT}, \quad f \in R/Z(1/T).$$

Se calcoliamo la relazione precedente per $f = kF$, dove $F = 1/NT$, otteniamo proprio

$$X(kF) = T \sum_{n=0}^{N-1} x(nT)e^{-j2\pi \frac{nk}{N}},$$

che ha la stessa struttura della (17). Dunque, N campioni in un periodo di $X(f)$, $X(kF)$, $k = 0, \dots, (N-1)F$, possono essere calcolati usando la FFT. Si noti che possiamo infittire i campioni in cui viene calcolata la trasformata $X(f)$ semplicemente aumentando la dimensione da N a $M > N$: in tale caso, i campioni aggiuntivi $x(NT), \dots, x((M-1)T)$ da usare per il calcolo usando la (17) sono evidentemente uguali a zero.

Se il segnale $x(nT)$ ha durata finita $D \leq NT$, ma la sua estensione non è contenuta nella cella elementare $\{0, \dots, (N-1)T\}$, occorre osservare che ai fini del calcolo dei *campioni* $X(kF)$, $k = 0, \dots, (N-1)F$, possiamo considerare la versione *periodicizzata* del segnale di ingresso

$$x_p(nT) = \text{per}_{NT}x(nT).$$

Infatti, se $x(nT)$ ha estensione contenuta in $\{n_0T, (n_0+1)T, \dots, (n_0+N-1)T\}$, si ha

$$X(kF) = T \sum_{n=n_0}^{n_0+N-1} x(nT)e^{-j2\pi nTkF} = T \sum_{n=0}^{N-1} x_p(nT)e^{-j2\pi \frac{nk}{N}}.$$

Nel calcolo dei campioni della trasformata tramite la (17) occorre dunque considerare i valori nella cella $\{0, \dots, (N-1)T\}$ del *segnale periodicizzato*.

Infine, se il segnale $x(nT)$ ha durata praticamente limitata, è possibile considerarlo nullo entro una precisione prefissata al di fuori di un certo intervallo I_0 . Questo equivale in effetti a moltiplicare il segnale di ingresso per una finestra che vale 1 per $t \in I_0$ e 0 altrove. I valori calcolati dei campioni della trasformata saranno pertanto relativi alla convoluzione tra la trasformata di Fourier del segnale di ingresso con quella della finestra.

Stima della trasformata per segnali a tempo continuo. Supponiamo di avere un segnale a tempo continuo $x(t), t \in R$ e di volere calcolare una stima della sua trasformata di Fourier $X(f), f \in R$. Per potere utilizzare a tale scopo un algoritmo numerico, e in particolare la FFT, occorre dapprima campionare il segnale di ingresso, ponendo $x_c(t) = x(t), t \in Z(T)$. La scelta di

T deve essere fatta in modo da limitare l'*aliasing*, e in generale deve essere $F_c = 1/T > B$, dove B è la larghezza di banda del segnale. Una volta campionato, il segnale ha una trasformata di Fourier periodica $X_c(f) = \text{rep}_{F_c} X(f)$, $f \in R/Z(F_c)$ i cui campioni in un periodo possono essere stimati mediante un algoritmo di FFT, secondo quanto esposto relativamente ai segnali a tempo discreto. Se non si conosce a priori la larghezza di banda B del segnale, occorrerà procedere per tentativi, valutando se le stime che si ottengono variano entro la precisione desiderata al variare di T .

Se il segnale $x(t)$ è reale, la sua trasformata di Fourier ha simmetria hermitiana ed estensione spettrale simmetrica $[-B/2, B/2]$. L'algoritmo di FFT fornisce i campioni della trasformata periodica $X_c(f) = \text{rep}_{F_c} X(f)$ relativamente al periodo $[0, F_c]$, mentre il confronto con $X(f)$ è più significativo se si considera la cella elementare $[-F_c/2, F_c/2]$. Posto $F = F_c/N$, una volta calcolati i valori $X_c(0), \dots, X_c((N-1)F)$ con un algoritmo di FFT, converrà dunque riferirsi al vettore

$$\left[X_c\left(\frac{N}{2}F\right), X_c\left(\left(\frac{N}{2} + 1\right)F\right), \dots, X_c((N-1)F), X_c(0), \dots, X_c\left(\left(\frac{N}{2} - 1\right)F\right) \right],$$

per N pari, e a

$$\left[X_c\left(\frac{N+1}{2}F\right), X_c\left(\left(\frac{N+1}{2} + 1\right)F\right), \dots, X_c((N-1)F), X_c(0), \dots, X_c\left(\left(\frac{N-1}{2}\right)F\right) \right],$$

per N dispari, che contengono la sequenza dei campioni relativi al periodo $[-F_c/2, F_c/2]$. Tale operazione di ordinamento degli elementi del vettore di uscita può ottenersi con la funzione MATLAB `fftshift`.

Esercizi

1. Si consideri il segnale $x(t), t \in Z(T)$, $T = 2$, con $x(0) = 1$, $x(\pm T) = 2$, $x(\pm 2T) = 3$ e $x(nT) = 0$ altrove. Si calcoli in forma chiusa la trasformata di Fourier $X(f), f \in R/Z(1/T)$. Si calcolino inoltre N campioni della trasformata in un periodo usando un algoritmo di FFT, per $N = 8, 16, 256$, e si verifichi in un grafico la coincidenza fra i campioni calcolati e l'espressione teorica.

Si valuti inoltre approssimativamente il numero di operazioni richieste per il calcolo degli N campioni della trasformata usando la FFT ed il calcolo diretto. Si tenga presente che il segnale di ingresso ha solamente 5 campioni diversi da zero.

Nota: $X(f)$ è in generale una funzione complessa, ed occorre considerarne di volta in volta la parte reale e immaginaria, oppure il modulo e la fase.

2. Si calcoli la DFT dei seguenti segnali, definiti su $Z(T)$, e si disegni il grafico del modulo, cercando di spiegarne le caratteristiche. Sia $F = 1/T = 8$ kHz.

a. Trasformata su $N = 512$ punti. Segnale di ingresso

$$x(nT) = 1, \quad 0 \leq n \leq 63 \\ 0, \quad \text{altrove.}$$

b. Trasformata su $N = 256$ punti. Segnale di ingresso

$$x(nT) = 0.5 \sin\left(\frac{2\pi n}{256} 9\right) + 0.7 \sin\left(\frac{2\pi n}{256} 100\right), \quad 0 \leq n \leq 255.$$

c. Trasformata su $N = 256$ punti. Segnale di ingresso

$$x(nT) = 0.5 \sin\left(\frac{2\pi n}{256} \frac{100}{3}\right) + 0.7 \sin\left(\frac{2\pi n}{256} \frac{200}{3}\right), \quad 0 \leq n \leq 255.$$

3. Una sinusoide di frequenza 50 Hz viene campionata con 500 campioni al secondo e analizzata usando una FFT su 64 campioni.

a. Quale campione della trasformata sarà maggiore in modulo?

b. Quale campione della trasformata avrà un'ampiezza immediatamente inferiore a quella del campione di modulo maggiore, e qual'è l'ampiezza relativa dei due campioni (esprimere il risultato in dB)?

Si verifichino i risultati con MATLAB.

4. Siano dati i seguenti segnali, definiti su $Z(T)$ per $T = 1$.

a. $x(nT) = \sin\left(\frac{2\pi n}{256} 172\right)$,

b. $x(nT) = \sin(n)$.

Si analizzino i due segnali mediante FFT su 256 punti. Si preveda quale sarà il coefficiente della trasformata più grande in modulo nei due casi, e l'ampiezza relativa di tali campioni della trasformata. Si verifichi il risultato con MATLAB.

5. Questo esercizio mostra come occorra fare attenzione nell'uso della FFT per il calcolo della trasformata di Fourier di segnali sinusoidali. Si consideri il segnale $x(t) = \cos 2\pi f_0 t$, $t \in Z(T)$, $T = 1$, $f_0 = 0.25$ e si costruisca il vettore MATLAB

$$\mathbf{x}(n+1) = x(nT), \quad n = 0, \dots, N-1$$

per $N = 32$ e $N = 64$. Si calcoli poi la FFT di \mathbf{x} nei due casi su N punti e la si disegni usando la funzione `stem`. Si discuta il risultato ottenuto. Si ripeta lo stesso esperimento

per $f_0 = 0.254$ e si discuta il risultato ottenuto. Si risolva inoltre l'esercizio per via analitica e si confronti con la soluzione trovata per via numerica.

Nota: per $f_0 = 0.25$ e i due valori di N considerati, $x(t)$ è periodico di periodo NT . Per $f_0 = 0.254$, $x(t)$ non è periodico di periodo NT .

6. Si scriva una procedura analoga alla procedura MATLAB `freqz`, usando la FFT.
7. Dati i segnali $x(nT)$ con estensione $0, \dots, (N-1)T$ e $h(nT)$ con estensione $0, \dots, (M-1)T$, si scriva una procedura MATLAB che calcola la convoluzione $y(nT) = h * x(nT)$ mediante l'uso della FFT. Si confronti il risultato con quello fornito dalla funzione MATLAB `y=T*conv(h,x)`, in cui si pone $\mathbf{x}(n+1) = x(nT)$, $nT = 0, \dots, (N-1)T$, e $\mathbf{h}(n+1) = h(nT)$, $nT = 0, \dots, (M-1)T$. Si scriva un programma che calcola la convoluzione di due segnali generici con estensioni $-5T, \dots, 4T$ e $3T, \dots, 5T$.

Soluzione proposta per la prima parte:

```
function y=convv(T,x,h);

N=length(x); M=length(h);

L=N+M-1;

X=T*fft(x,L); H=T*fft(h,L);

Y=X.*H;

y=ifft(Y)/T;

% Se x e h sono reali, y e' reale. Si pu\o eliminare
% l'eventuale parte immaginaria, non identicamente nulla a causa
% degli errori di calcolo.

if (isreal(x) & isreal(h)) y=real(y);

end;
```

8. *Metodo Overlap-Add.* Con riferimento all'esercizio precedente, se il segnale $x(t)$, $t \in Z(T)$ ha una durata NT molto più grande di quella MT di $h(t)$, non è efficiente calcolare direttamente la convoluzione usando la FFT, che deve essere valutata per entrambi i

segnali su almeno $M + N - 1$ punti. È conveniente allora suddividere il segnale di ingresso in blocchi non sovrapposti lunghi L ,

$$x_i(nT) = \begin{cases} x(nT), & n = iL, \dots, iL + L - 1, \\ 0, & \text{altrove} \end{cases}$$

ponendo

$$x(nT) = \sum_i x_i(nT).$$

Il risultato della convoluzione viene ottenuto *sommando* le uscite $x_i * h(nT)$, ciascuna delle quali può essere calcolata usando la FFT. Si noti che le uscite $x_i * h(nT)$ risultano sovrapposte di $M - 1$ campioni.

Si scriva un programma MATLAB che calcola la convoluzione di due segnali con il metodo *Overlap-Add*, ponendo $M = 4$ e $N = 64$, e si confronti il risultato con il metodo diretto e quello dell'esercizio 7.

9. Dati due segnali $x_1(t)$ e $x_2(t)$, $t \in Z(T)$ a durata limitata LT , è possibile considerarne le ripetizioni periodiche $x_{1,p}(t) = \text{rep}_{LT}x_1(t)$, $t \in Z(T)/Z(LT)$ e $x_{2,p}(t) = \text{rep}_{LT}x_2(t)$, $t \in Z(T)/Z(LT)$, senza perdita di informazione. Dimostrare e verificare su un esempio sviluppato con MATLAB che la convoluzione *ciclica* $y_p(nT) = x_{1,p} * x_{2,p}(nT)$ è uguale alla ripetizione periodica della convoluzione $y(nT) = x_1 * x_2(nT)$, ovvero $y_p(nT) = \text{rep}_{LT}y(nT)$.
10. *Metodo Overlap-Save*. Si considerino, come nell'Esercizio 8, due segnali $x(t)$, $t \in Z(T)$ e $h(t)$ con estensione $[0, \dots, (N - 1)T]$ e $[0, \dots, (M - 1)T]$, rispettivamente, e $N \gg M$. Nel metodo *Overlap-Save* per il calcolo della convoluzione, il segnale $x(nT)$ viene suddiviso in blocchi $x_k(nT)$ lunghi L campioni. Il metodo consiste nel calcolare una convoluzione ciclica (vedi Esercizio 9) tra $h(nT)$ e $x_k(nT)$, identificando quella parte della convoluzione ciclica che corrisponde alla normale convoluzione. In particolare, supponendo $L \geq M$, nella convoluzione ciclica di $h(nT)$ e $x_k(nT)$ calcolata su L punti, risulta che i primi $M - 1$ non sono corretti, mentre i rimanenti punti sono gli stessi che otterremmo dalla normale convoluzione (perchè? Vedi l'Esercizio 9). Conviene dunque sezionare $x(nT)$ in segmenti di lunghezza L in modo che ogni segmento si sovrapponga al precedente per $M - 1$ punti. Dopo avere definito i segmenti $x_k(nT)$ nel modo seguente

$$x_k(nT) = x((n + k(L - M + 1))T), \quad 0 \leq n \leq L - 1,$$

si calcola pertanto la convoluzione ciclica $y_k(nT) = h * x_k(nT)$ su L punti e se ne scartano i primi $M - 1$. I rimanenti punti di ogni sottosequenza $y_k(nT)$ vengono giustapposti gli uni di seguito agli altri, fino ad ottenere l'uscita filtrata finale.

Si scriva un programma MATLAB che calcola la convoluzione di due segnali con il metodo *Overlap-Save*, ponendo $M = 4$ e $N = 64$, e si confronti il risultato con il metodo diretto e quello dell'Esercizio 7.

11. Si stimi la trasformata di Fourier del segnale a tempo continuo

$$x(t) = e^{-|6t|}, \quad t \in R.$$

12. Si stimi la trasformata di Fourier del segnale a tempo continuo

$$x(t) = e^{-|6(t-1)|}, \quad t \in R.$$

13. Si stimi la trasformata di Fourier del segnale a tempo continuo

$$x(t) = e^{-|6t|} \cos(2\pi 0.5t), \quad t \in R.$$

14. Si stimi la trasformata di Fourier del segnale a tempo continuo

$$x(t) = \frac{\sin(\pi(1-r)t/T) + 4r(t/T) \cos(\pi(1+r)t/T)}{\pi[1 - (4rt/T)^2]} \text{rect} \frac{t}{8T}, \quad r = 0.125, \quad t \in R.$$