

Trasformata  $z$  e disposizione di poli e zeri  
nei filtri FIR e IIR

Walter Mottinelli e Andrea Bo

29 giugno 2004

# Indice

<b>1</b>	<b>La trasformata <math>z</math></b>	<b>1</b>
1.1	Definizione matematica . . . . .	1
1.2	Proprietà della trasformata $z$ . . . . .	2
1.3	Trasformate $z$ comuni . . . . .	3
1.4	Poli e zeri . . . . .	3
1.5	La regione di convergenza (ROC) . . . . .	6
1.5.1	Determinazione della ROC di un segnale causale . . .	7
1.5.2	Determinazione della ROC di un segnale non causale .	7
1.5.3	Determinazione della ROC di un segnale bilatero . . .	9
1.6	Calcolo della trasformata inversa . . . . .	11
1.6.1	Metodo delle funzioni parziali . . . . .	11
1.6.2	Metodo della divisione . . . . .	12
1.6.3	Metodo del prodotto di Cauchy . . . . .	13
1.6.4	Metodo di integrazione . . . . .	13
<b>2</b>	<b>Trasformata <math>z</math> e filtri digitali</b>	<b>15</b>
2.1	Calcolo della funzione di trasferimento... . . . .	15
2.2	Disposizione di poli e zeri in un filtro generico . . . . .	17
2.2.1	Grafico della risposta in frequenza e in fase . . . . .	18
2.3	Progetto di filtri mediante disposizione iterativa di poli/zeri . . . . .	20
2.3.1	Realizzazione iterativa di un filtro passa-basso . . . .	21
<b>3</b>	<b>Disposizione di poli e zeri nei filtri FIR</b>	<b>29</b>
3.1	Filtro con uno zero sulla circonferenza unitaria . . . . .	29
3.2	Filtri COMB . . . . .	30
3.3	Filtri realizzati con il metodo delle finestre . . . . .	33
3.4	Filtro passa-alto realizzato con il metodo minimax . . . . .	36
<b>4</b>	<b>Disposizione di poli e zeri nei filtri IIR</b>	<b>40</b>
4.1	Filtri di Butterworth . . . . .	40
4.2	Filtri di Chebyshev . . . . .	41
4.2.1	Filtro di Chebyshev - tipo I . . . . .	41

4.2.2	Filtro di Chebyshev - tipo II . . . . .	43
4.3	Filtri ellittici . . . . .	44
<b>A</b>	<b>Introduzione a Scilab</b>	<b>48</b>
A.1	I fondamenti . . . . .	48
A.2	La riga di comando . . . . .	48
A.3	Vettori e matrici . . . . .	48
A.4	Visualizzazione dei grafici . . . . .	50
A.5	Salvataggio e ripristino dell'ambiente di lavoro . . . . .	51
A.6	Programmazione . . . . .	51
A.7	Polinomi e funzioni di trasferimento . . . . .	51
A.8	Funzioni per l'analisi numerica dei segnali . . . . .	52
<b>B</b>	<b>Introduzione a Scicos</b>	<b>54</b>
B.1	L'editor di Scicos . . . . .	54
B.2	Blocchi base . . . . .	55
B.3	Impostazione dei parametri delle componenti . . . . .	55
B.4	Simulazione . . . . .	55
B.5	Altre funzionalità . . . . .	56
B.6	Esempio: simulazione di un filtro passa-basso . . . . .	56

## **Sommario**

In questo documento viene presentata la trasformata  $z$  e la disposizione di poli e zeri nel piano complesso. In relazione all'analisi numerica dei segnali, viene mostrata la correlazione tra il grafico poli/zeri di una funzione di trasferimento di un filtro e la risposta all'impulso dello stesso, con alcuni esempi di filtri FIR e IIR di uso comune.

Gli esempi e le illustrazioni che accompagnano la trattazione sono realizzati in Scilab, un applicativo free per il calcolo numerico.

# Capitolo 1

## La trasformata $z$

La trasformata  $z$  è uno dei principali strumenti matematici per l'analisi e la sintesi di filtri digitali. Trova infatti applicazione in alcune importanti problematiche, ad esempio:

- *Calcolo della funzione di trasferimento del sistema*

La funzione di trasferimento di un filtro è la trasformata  $z$  della sua risposta all'impulso.

Il piano complesso  $z$  è utilizzato per mostrare la configurazione dei poli e degli zeri della funzione di trasferimento: la loro disposizione, come vedremo, è indice di importanti proprietà del sistema.

Inoltre, i coefficienti del filtro digitale possono essere ricavati direttamente a partire dalla funzione di trasferimento.

- *Calcolo delle trasformate di Fourier*

La trasformata di Fourier corrisponde al calcolo della trasformata  $z$  lungo la circonferenza di raggio unitario con centro nell'origine degli assi del piano complesso  $z$ . In particolare, la versione discreta della trasformata di Fourier viene valutata su insiemi di punti equidistanti tra loro.

### 1.1 Definizione matematica

La trasformata  $z$  di una sequenza  $\{h[n]\}$  è definita dalla funzione nella variabile  $z$  a valori complessi

$$H(z) = \sum_{n=-\infty}^{\infty} h(n)z^{-n}$$

e dalla regione di convergenza, luogo dei punti in cui la sommatoria è finita. Se la sequenza  $\{h[n]\}$  in esame è la risposta all'impulso di un filtro digitale,  $H(n)$  viene chiamata *funzione di trasferimento del sistema*.

**Esempio di calcolo della trasformata  $z$** 

Sia data la funzione

$$h(n) = r^n \cos(\omega_0 n)$$

per  $n \geq 0$ , la cui trasformata  $z$  è uguale a

$$H(z) = \sum_{n=0}^{\infty} r^n \cos(\omega_0 n) z^{-n}$$

Applicando la formula

$$\cos \omega = \frac{e^{i\omega} + e^{-i\omega}}{2}$$

derivata dall'identità di Eulero  $e^{i\omega} = \cos \omega + i \sin \omega$  e la formula della somma infinita di elementi di una serie geometrica

$$\sum_{n=0}^{\infty} a^n = \frac{1}{1-a} \text{ per } |a| < 1$$

otteniamo:

$$\begin{aligned} H(z) &= \sum_{n=0}^{\infty} r^n \frac{e^{i\omega_0 n} + e^{-i\omega_0 n}}{2} z^{-n} \\ &= \frac{1}{2} \left[ \sum_{n=0}^{\infty} (r e^{i\omega_0} z^{-1})^n + \sum_{n=0}^{\infty} (r e^{-i\omega_0} z^{-1})^n \right] \\ &= \frac{1}{2} \left[ \frac{1}{1 - r e^{i\omega_0} z^{-1}} + \frac{1}{1 - r e^{-i\omega_0} z^{-1}} \right] \\ &= \frac{1}{2} \left[ \frac{2 - r(e^{i\omega_0} + e^{-i\omega_0} z^{-1})}{(1 - r e^{i\omega_0} z^{-1})(1 - r e^{-i\omega_0} z^{-1})} \right] \\ &= \frac{1 - r \cos(\omega_0) z^{-1}}{1 - 2r \cos(\omega_0) z^{-1} + r^2 z^{-2}} \end{aligned}$$

**1.2 Proprietà della trasformata  $z$** 

- linearità  
se  $\{x(n)\} = \{a h_1(n) + b h_2(n)\}$ , allora  $X(z) = a H_1(z) + b H_2(z)$
- traslazione  
se  $\{y(n)\} = \{x(n - n_0)\}$ , allora  $Y(z) = z^{-n_0} X(z)$
- convoluzione  
se  $\{y(n)\} = \{h(n)\} * \{x(n)\}$ , allora  $Y(z) = H(z) X(z)$
- scalatura (nel dominio  $z$ )  
se  $\{y(n)\} = \{a^n x(n)\}$ , allora  $Y(z) = X(a^{-1} z)$
- derivazione (nel dominio  $z$ )  
se  $\{y(n)\} = \{n x(n)\}$ , allora  $Y(z) = z^{-1} \frac{dX}{dz^{-1}}$

### 1.3 Trasformate $z$ comuni

Segnale	Trasformata $z$	ROC
$\delta[n]$	1	$\forall z$
$\delta[n-k]$	$z^{-k}$	$\forall z$ , con $z \neq 0$ se $k > 0$
$\delta[n-k]$	$z^{-k}$	$\forall z$ , con $z \neq \infty$ se $k < 0$
$u[n]$	$\frac{z}{z-1}$	$ z  > 1$
$-(u[-n-1])$	$\frac{z}{z-1}$	$ z  < 1$
$nu[n]$	$\frac{z}{(z-1)^2}$	$ z  > 1$
$n^2u[n]$	$\frac{z(z+1)}{(z-1)^3}$	$ z  > 1$
$n^3u[n]$	$\frac{z(z^2+4z+1)}{(z-1)^4}$	$ z  > 1$
$(-\alpha^n)u[-n-1]$	$\frac{z}{z-\alpha}$	$ z  <  \alpha $
$\alpha^n u[n]$	$\frac{z}{z-\alpha}$	$ z  >  \alpha $
$n\alpha^n u[n]$	$\frac{\alpha z}{(z-\alpha)^2}$	$ z  >  \alpha $
$n^2\alpha^n u[n]$	$\frac{\alpha z(z+\alpha)}{(z-\alpha)^3}$	$ z  >  \alpha $
$\frac{\prod_{k=1}^m (n-k+1)}{\alpha^m m!} \alpha^n u[n]$	$\frac{z}{(z-\alpha)^{m+1}}$	
$\gamma^n \cos(\alpha n) u[n]$	$\frac{z(z-\gamma \cos(\alpha))}{z^2 - (2\gamma \cos(\alpha))z + \gamma^2}$	$ z  >  \alpha $
$\gamma^n \sin(\alpha n) u[n]$	$\frac{z\gamma \sin(\alpha)}{z^2 - (2\gamma \cos(\alpha))z + \gamma^2}$	$ z  >  \alpha $

### 1.4 Poli e zeri

La trasformata  $H(z)$  è una funzione nella variabile a valori complessi  $z$ , che può essere rappresentata su un piano con la parte reale sull'asse delle ascisse e la parte immaginaria sull'asse delle ordinate. Alcuni punti del piano complesso hanno particolare importanza:

- gli *zeri* sono i punti in cui  $H(z) = 0$
- i *poli* sono i punti in cui  $H(z) = \infty$

Con un grafico tridimensionale è possibile evidenziare il comportamento della trasformata in questi punti: in figura 1.1 è rappresentato in funzione di  $\Re(z)$  e  $\Im(z)$  il modulo di  $H(z) = z^{-1} - 1$ , che ha uno zero in  $z = 1$  (l'avvallamento) e un polo in  $z = 0$  (il picco).

Poli e zeri possono essere indicati esplicitamente nell'equazione di  $H(z)$ : consideriamo ad esempio una funzione di trasferimento di un segnale digitale, normalmente rappresentata come rapporto di somme di prodotti

$$H(z) = \frac{\sum_{k=-N_F}^{N_P} b_k z^{-k}}{1 - \sum_{k=1}^M a_k z^{-k}}$$

dove

- $N_F$  è il numero dei campioni in input successivi a quello in esame;

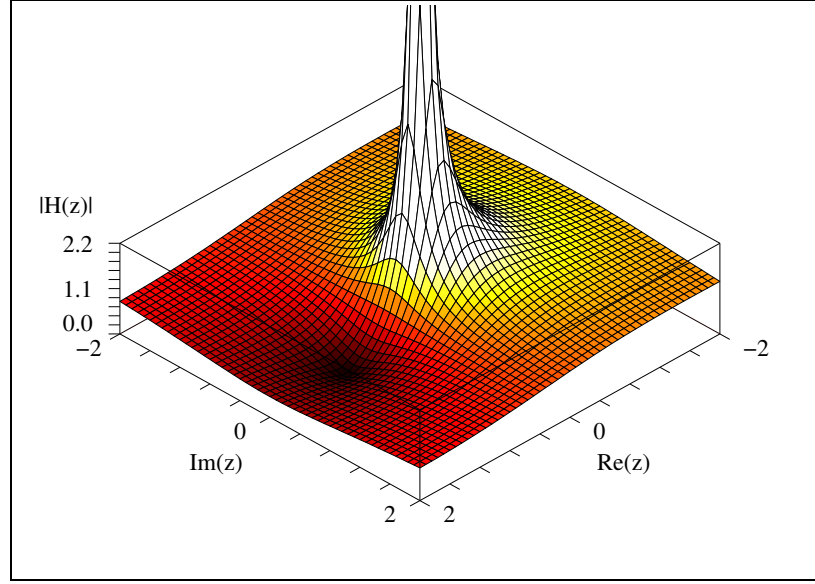


Figura 1.1: Modulo di  $H(z) = z^{-1} - 1$  in funzione di  $\Re(z)$  e  $\Im(z)$

- $N_P$  è il numero dei campioni in input precedenti quello in esame;
- $M$  è il numero dei campioni in output precedenti quello in esame.

A partire da questa equazione è possibile passare ad un rapporto di prodotti di termini

$$H(z) = Az^{N_F} \frac{\prod_{k=1}^{N_P+N_F} (1 - c_k z^{-1})}{\prod_{k=1}^M (1 - d_k z^{-1})}$$

dove

- $A$  è una costante reale;
- $c_k$ , per  $1 \leq k \leq N_P + N_F$ , è il valore (eventualmente complesso) del  $k$ -esimo zero;
- $d_k$ , per  $1 \leq k \leq M$ , è il valore del  $k$ -esimo polo.

Alcune considerazioni sulla seconda rappresentazione:

- ogni fattore del numeratore  $(1 - c_k z^{-1})$  genera uno zero in  $z = c_k$  e un polo in  $z = 0$ ;
- ogni fattore del denominatore genera un polo in  $z = d_k$  e uno zero in  $z = 0$ ;
- nel punto  $z = 0$ , poli dei fattori del numeratore cancellano zeri dei fattori del denominatore;



- considerando anche i punti di singolarità in  $z = \infty$ , il numero dei poli è *sempre* pari al numero degli zeri.

In sintesi, la disposizione dei poli e degli zeri della trasformata  $z$  data nella forma

$$H(z) = Az^{N_F} \frac{\prod_{k=1}^{N_P+N_F} (1 - c_k z^{-1})}{\prod_{k=1}^M (1 - d_k z^{-1})}$$

è la seguente:

- zeri in:
  - $z = c_k$  per  $1 \leq k \leq N_P + N_F$
  - $z = 0$  di molteplicità  $M$
  - $z = 0$  di molteplicità  $N_F$
- poli in:
  - $z = d_k$  per  $1 \leq k \leq M$
  - $z = 0$  di molteplicità  $N_P + N_F$
  - $z = \infty$  di molteplicità  $N_F$

### Esempio 1

$$\begin{aligned} H(z) &= \frac{1 - r \cos(\omega_0) z^{-1}}{1 - 2r \cos \omega_0 z^{-1} + r^2 z^{-2}} \\ &= \frac{1 - r \cos(\omega_0) z^{-1}}{1 - r(e^{i\omega_0} + e^{-i\omega_0}) z^{-1} + r^2 z^{-2}} \\ &= \frac{1 - r \cos(\omega_0) z^{-1}}{(1 - r e^{i\omega_0} z^{-1})(1 - r e^{-i\omega_0} z^{-1})} \end{aligned}$$

Il numeratore genera uno zero in  $z = r \cos(\omega_0)$  e un polo in  $z = 0$ , mentre il denominatore genera due zeri in  $z = 0$  e un polo in  $z = r e^{\pm i\omega_0}$ .

### Esempio 2 - singolarità di second'ordine

$$\begin{aligned} H(z) &= z^2 + 2z + 3 + 2z^{-1} + z^{-2} \\ &= (z + 1 + z^{-1})^2 \\ &= z^2 (1 - c_1 z^{-1})^2 (1 - c_2 z^{-1})^2 \end{aligned}$$

dove

$$\begin{aligned} c_1 &= \frac{-1 + i\sqrt{3}}{2} = e^{-i2\pi/3} \\ c_2 &= \frac{-1 - i\sqrt{3}}{2} = e^{-i2\pi/3} \end{aligned}$$

Il primo fattore genera due poli in  $z = \infty$  e due zeri in  $z = 0$ . I due fattori contenenti  $z^{-1}$  generano ciascuno due zeri in  $z = c_1$  e  $z = c_2$  e due poli in  $z = 0$ .

### Esempio 3 - zeri disposti sulla circonferenza unitaria

$H(z) = 1 - z^{-N}$  é un polinomio di ordine  $N$ , e ha radici

$$\begin{aligned} z^N &= r e^{i(\theta+2m\pi)/N} \\ &= r^{1/N} e^{i(\theta+2m\pi)/N} \end{aligned}$$

dove  $m$  è un intero. Vale

$$z^N = 1 = e^{i2m\pi}$$

quindi gli zeri sono in  $z = e^{i2m\pi/N}$ , per  $m = 0, 1, 2, \dots, N-1$ : la loro posizione è lungo la circonferenza di raggio unitario e centro nell'origine degli assi, a partire da  $z = 1$  e distanziati di un angolo  $2\pi/N$ .

## 1.5 La regione di convergenza (ROC)

La trasformata  $z$  è definita solo per valori di  $z$  in cui  $\sum_{n=-\infty}^{\infty} |h(n)z^{-n}| < \infty$ . Poiché  $H(z)$  va all'infinito nei punti in cui sono definiti i poli, questi non possono trovarsi nella ROC.

Per individuare la ROC è possibile procedere nel seguente modo:

- per i segnali di durata finita, la ROC è l'intero piano  $Z$  escluso
  - $z = 0$  per segnali causali
  - $z = \infty$  per segnali non causali
  - $z = 0$  e  $z = \infty$  per i segnali bilaterali
- per i segnali di durata infinita, la ROC è
  - nel caso di segnali causali, l'intero piano eccetto il cerchio di raggio  $r_1$  pari alla distanza tra l'origine e il polo finito più lontano
  - nel caso di segnali non causali, il cerchio di raggio  $r_2$  pari alla distanza tra l'origine e il polo finito più vicino
  - nel caso di segnali bilaterali, l'anello racchiuso dalle circonferenze di raggio  $r_1$  e  $r_2$

La ROC è indispensabile per ottenere il segnale nel dominio temporale a partire dalla sua trasformata  $z$ , in quanto non vi è corrispondenza biunivoca tra un segnale e la sola forma analitica della sua trasformata  $z$ : segnali diversi possono avere la stessa trasformata, e quindi gli stessi poli e zeri, ma regione di convergenza diversa, come mostrato nel prossimo paragrafo.

### 1.5.1 Determinazione della ROC di un segnale causale

Prendiamo in considerazione il segnale

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

con

$$u(n) = \begin{cases} 1 & \text{per } n \geq 0 \\ 0 & \text{altrimenti} \end{cases}$$

cioè l'analogo discreto del segnale gradino unitario.

Calcolando la trasformata  $z$  del segnale  $x(n)$  si ricava

$$X(z) = \sum_{n=-\infty}^{\infty} a^n u(n) z^{-n} = \frac{1}{1 - az^{-1}} = \frac{z}{z - a}$$

Analizzando la trasformata  $z$  si rileva uno zero in  $z = 0$  e un polo in  $z = a$ : la regione di convergenza è così definita per  $|z| > |a|$ , essendo il segnale causale. Il diagramma poli/zeri può essere ottenuto con Scilab tramite i seguenti comandi:

```
// zero della funzione di trasferimento del filtro
a=.5;
// creazione del numeratore
Ncoeff=[0,1];
N=poly(Ncoeff,'z','coeff');
// creazione del denominatore
Dcoeff=[-(a),1];
D=poly(Dcoeff,'z','coeff');
// creazione della funzione di trasferimento H(z)=N/D
Ztrasf=syslin('d',N,D);
// disegno del grafico dei poli/zeri
xbasc(); xset("font size",4); plzr(Ztrasf);
// disegno della circonferenza che delimita la ROC (r=1)
xarc(-0.5,0.5,1,1,0,360*64)
```

Come risultato otteniamo il grafico rappresentato in figura 1.2, in cui la ROC è stata evidenziata con una colorazione più scura.

### 1.5.2 Determinazione della ROC di un segnale non causale

Prendiamo in considerazione il segnale

$$x(n) = -a^n u(-n - 1)$$

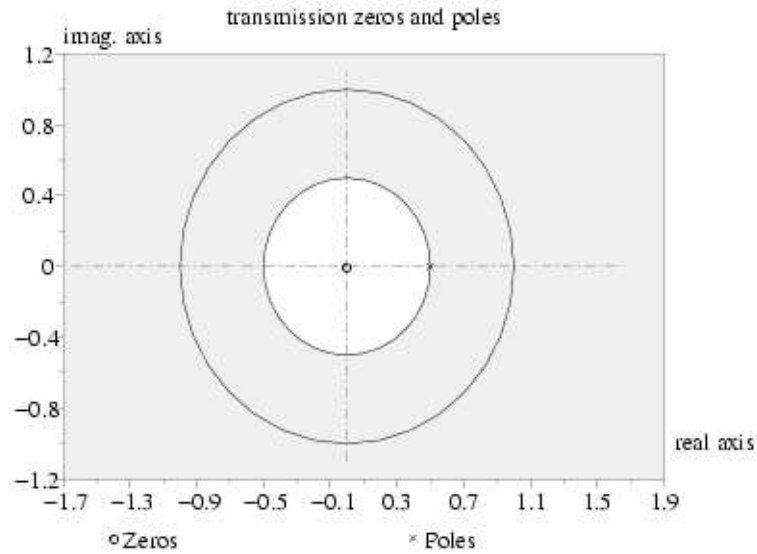


Figura 1.2: Grafico della ROC e dei poli/zeri di un segnale causale

Calcolando la trasformata  $z$  del segnale  $x(n)$  si ricava

$$\begin{aligned}
 X(z) &= - \sum_{n=-\infty}^{\infty} a^n u(-n-1) z^{-n} \\
 &= - \sum_{n=-\infty}^{-1} a^n z^{-n} \\
 &= 1 - \sum_{n=0}^{\infty} (a^1 z)^{-n} \\
 &= \frac{z}{z-a}
 \end{aligned}$$

Analizzando la trasformata  $z$  si rileva uno zero in  $z = 0$  e un polo in  $z = a$ : la regione di convergenza è definita per  $|z| < |a|$ , essendo il segnale non causale.

Il diagramma poli/zeri può essere ottenuto con Scilab tramite i seguenti comandi:

```
// zero della funzione di trasferimento del filtro
a=.5;
// creazione del numeratore
Ncoeff=[0,1];
N=poly(Ncoeff,'z','coeff');
// creazione del denominatore
Dcoeff=[-(a),1];
```

```

D=poly(Dcoeff,'z','coeff');
// creazione della funzione di trasferimento H(z)=N/D
Ztrasf=syslin('d',N,D);
// disegno del grafico dei poli/zeri
xbasc(); xset("font size",4); plzr(Ztrasf);
// disegno della circonferenza che delimita la ROC (r=1)
xarc(-0.5,0.5,1,1,0,360*64)

```

Come risultato otteniamo il grafico rappresentato in figura 1.3, in cui la ROC è stata evidenziata con una colorazione più scura.

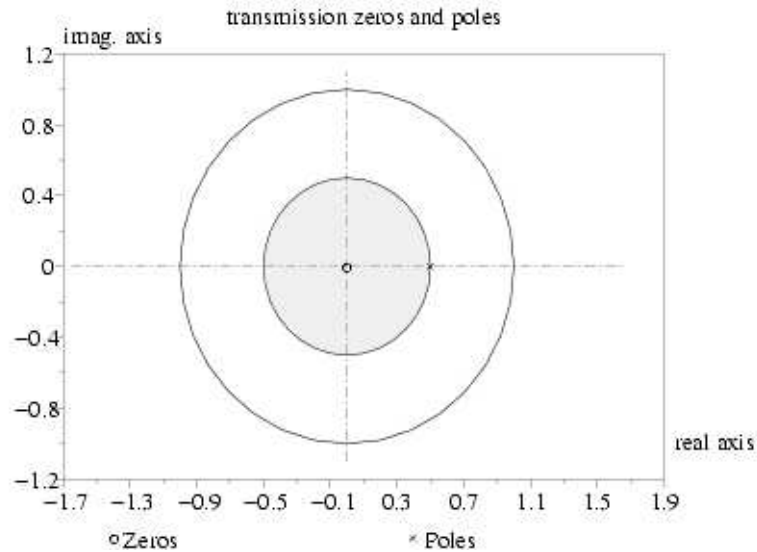


Figura 1.3: Grafico della ROC e dei poli/zeri di un segnale non causale

### 1.5.3 Determinazione della ROC di un segnale bilatero

Prendiamo in considerazione il segnale

$$x(n) = \left(-\frac{1}{4}\right)^n u(n) - \left(\frac{1}{2}\right)^n u(-n-1)$$

Calcoliamo la trasformata  $z$  del segnale  $x(n)$  scomponendolo in due parti:

$$\begin{aligned} \left(-\frac{1}{4}\right)^n u(n) &\longrightarrow \frac{z}{z + \frac{1}{4}} \\ -\left(\frac{1}{2}\right)^n u(n) &\longrightarrow \frac{z}{z - \frac{1}{2}} \end{aligned}$$

quindi la trasformata  $z$  finale risulta essere

$$X(z) = \frac{z}{z + \frac{1}{4}} + \frac{z}{z - \frac{1}{2}} = \frac{z(2z - \frac{1}{8})}{(z + \frac{1}{4})(z - \frac{1}{2})}$$

Analizzando la trasformata  $z$  si rilevano due zeri ( $z = 0$  e  $z = \frac{1}{16}$ ) e due poli ( $z = -\frac{1}{4}$  e  $z = \frac{1}{2}$ ).

Utilizzando la proprietà di linearità della trasformata  $z$  possiamo concludere che la regione di convergenza è definita come l'intersezione delle regioni di convergenza delle due parti del segnale, quindi è uguale a  $\frac{1}{4} < |z| < \frac{1}{2}$ .

Il diagramma dei poli-zeri può essere ottenuto con Scilab tramite i seguenti comandi:

```
// creazione del numeratore
Ncoeff=[0,-(1/8),2];
N=poly(Ncoeff,'z','coeff');
// creazione del denominatore
Dcoeff=[-(1/8),-(1/4),1];
D=poly(Dcoeff,'z','coeff');
// creazione della trasformata Z
Ztrasf=syslin('d',N,D)
// stampa del grafico
xbasc(); xset("font size",4); plzr(Ztrasf);
// disegno della circonferenza che delimita la ROC
xarc(-.5,.5,1,1,0,360*64) // Anello esterno
xarc(-.25,.25,.5,.5,0,360*64) // Anello interno
```

Come risultato otteniamo il grafico rappresentato in figura 1.4, in cui la ROC è stata evidenziata con una colorazione più scura.

Il diagramma del modulo e della fase del filtro può essere ottenuto con Scilab tramite i seguenti comandi:

```
// creazione del numeratore
Ncoeff=[0,-(1/8),2];
N=poly(Ncoeff,'z','coeff');
// creazione del denominatore
Dcoeff=[-(1/8),-(1/4),1];
D=poly(Dcoeff,'z','coeff');
// range di frequenze
f=(0:.01:.5);
// calcolo della risposta in frequenza
hf=freq(N,D,exp(2*pi*i*f));
xbasc(); xset("font size",4);
xsetech([0,0,1,.5]); plot2d(f,(abs(hf)))
xtitle("Risposta in frequenza del filtro")
```

```
xsetech([0,.5,1,.5]); plot2d(f,atan(imag(hf),real(hf)));
xtitle("Risposta in fase del filtro")
```

Come risultato otteniamo il grafico rappresentato in figura 1.5.

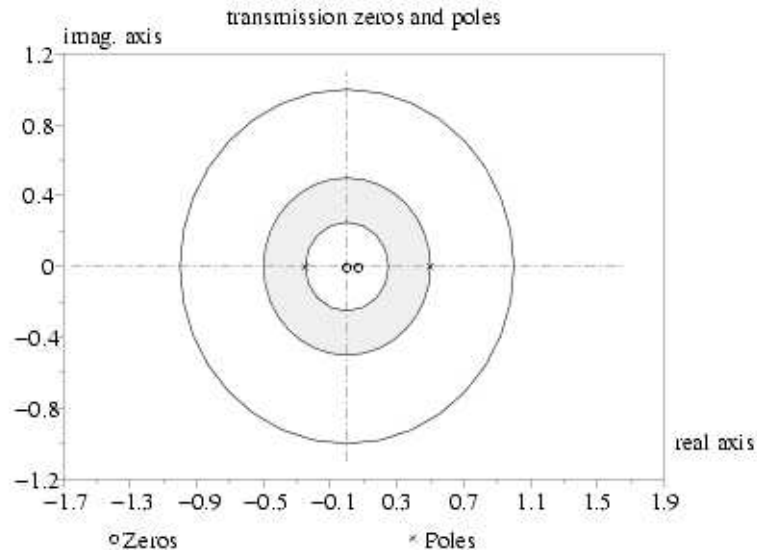


Figura 1.4: Grafico della ROC e dei poli/zeri di un segnale bilatero

## 1.6 Calcolo della trasformata inversa

La trasformata  $z$  inversa permette di tornare dal piano  $z$  al dominio d'origine: esistono quattro principali metodi per il calcolo dell'antitrasformata.

### 1.6.1 Metodo delle funzioni parziali

Uno dei metodi più semplici, e al tempo stesso più utili, per calcolare l'antitrasformata  $z$  consiste nel decomporre la trasformata in frazioni parziali. Consideriamo ad esempio la trasformata

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

Fattorizziamo il denominatore in modo da ottenere

$$X(z) = \frac{1}{(1 - z^{-1})(1 - \frac{1}{2}z^{-1})}$$

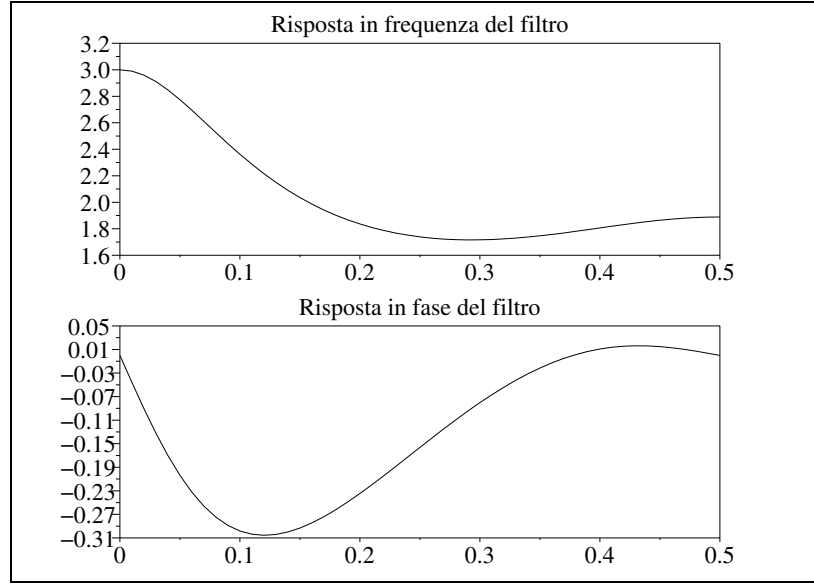


Figura 1.5: Risposta in frequenza e in fase di un filtro bilatero

$X(z)$  può quindi essere espressa come

$$X(z) = \frac{A}{1 - z^{-1}} + \frac{B}{1 - \frac{1}{2}z^{-1}}$$

con  $A = 2$  e  $B = -1$ .

Grazie alla proprietà di linearità, le due frazioni possono essere invertite separatamente: sono due trasformate semplici (presenti nella tabella delle trasformata più comuni), quindi otteniamo

$$x(n) = Z^{-1}X(z) = 2u(n) - \left(\frac{1}{2}\right)^n u(n)$$

### 1.6.2 Metodo della divisione

A partire dalla funzione di trasferimento espressa come rapporto tra numeratore e denominatore, è possibile eseguire la divisione polinomiale ed ottenere quindi un unico polinomio: la sua trasformata inversa è una successione di impulsi pesati. Riprendiamo l'esempio precedente:

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

Effettuando la divisione si ottiene:

$$X(z) = 1 + \frac{3}{2}z^{-1} + \frac{7}{4}z^{-2} + \frac{15}{8}z^{-3} + \dots$$



la cui trasformata inversa è

$$x(n) = \delta(n) + \frac{3}{2}\delta(n-1) + \frac{7}{4}\delta(n-2) + \frac{15}{8}\delta(n-3) + \dots$$

### 1.6.3 Metodo del prodotto di Cauchy

Analogamente al precedente, questo metodo consente di ottenere la trasformata inversa come somma di impulsi pesati e ritardati nel tempo. Se  $X(z)$  è nella forma

$$X(z) = \frac{b_0 + b_1 z^{-1} + \dots + b_m z^{-m}}{a_0 + a_1 z^{-1} + \dots + a_m z^{-m}}$$

applicando il prodotto di Cauchy (il prodotto di convoluzione) e manipolando l'equazione in  $x(n)$  si ottiene l'equazione ricorsiva

$$x(n) = \frac{1}{a_0} \left[ b_n - \sum_{k=0}^{n-1} x(k) a_{n-k} \right]$$

con  $a_0 \neq 0$ .

Nell'esempio precedente

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

i pesi sono

$$b_0 = 1, a_0 = 1, a_1 = -\frac{3}{2}, a_2 = \frac{1}{2}$$

È possibile applicare l'equazione ricorsiva in  $x(n)$  e ottenere

$$x(0) = 1, x(1) = \frac{3}{2}, x(2) = \frac{7}{4}, x(3) = \frac{15}{8}, \dots$$

### 1.6.4 Metodo di integrazione

Questo metodo è analiticamente il più complicato. È possibile dimostrare che esiste una relazione diretta tra l' $n$ -esimo elemento di una sequenza a tempo discreto  $h(n)$  e la sua trasformata  $z$ , secondo la relazione:

$$x(n) = \frac{1}{2\pi i} \oint_C X(z) z^{n-1} dz$$

dove l'integrazione è eseguita in senso antiorario lungo un percorso chiuso racchiudente l'origine degli assi ed interamente contenuto nella ROC.

Il calcolo diretto dell'integrale è spesso difficile, quindi viene utilizzato un secondo risultato, il *teorema dei residui*:

$$h(n) = \sum_{i=1}^N \text{Res}[H(z) z^{n-1} \text{ nel polo } p_i]$$

dove  $\text{Res}[\cdot]$  è il *residuo*. Per definire il residuo, sia  $H(z)z^{n-1}$  una funzione razionale in  $z$  con un polo di ordine  $m$  in  $z = z_0$ :

$$H(z)z^{n-1} = \frac{P(z)}{(z - z_0)^m}$$

Il residuo di  $H(z)z^{n-1}$  in  $z = z_0$  è definito come

$$\text{Res}[H(z)z^{n-1} \text{ nel polo } p_i] = \frac{1}{(m-1)!} \left. \frac{d^{m-1}P(z)}{dz^{m-1}} \right|_{z=z_0}$$

## Capitolo 2

# Trasformata $z$ e filtri digitali

### 2.1 Calcolo della funzione di trasferimento...

Nel dominio  $z$ , un sistema si comporta secondo la sua *funzione di trasferimento*, cioè la trasformata  $z$  della sua risposta all'impulso, quindi è importante poterla calcolare a partire dalle sequenze di ingresso e uscita o dalla equazione alle differenze.

#### ... date le sequenze di ingresso e di uscita

Consideriamo ad esempio il filtro che converte la sequenza di ingresso

$$x(n) = r^n \cos(\omega_0 n)$$

(per  $n \geq 0$ ) in

$$y(n) = r^n \sin(\omega_0 n)$$

(per  $n \geq 0$ ) e le loro trasformate

$$\begin{aligned} X(Z) &= \frac{1 - r \cos(\omega_0) z^{-1}}{1 - 2r \cos(\omega_0) z^{-1} + r^2 z^{-2}} \\ Y(Z) &= \frac{r \sin(\omega_0) z^{-1}}{1 - 2r \cos(\omega_0) z^{-1} + r^2 z^{-2}} \end{aligned}$$

La funzione di trasferimento richiesta è data da

$$\begin{aligned} H(z) &= Y(z)/X(z) \\ &= \frac{r \sin(\omega_0) z^{-1}}{1 - r \cos(\omega_0) z^{-1}} \end{aligned}$$

**... data l'equazione alle differenze**

Un sistema lineare tempo-invariante è descritto dalla seguente equazione alle differenze

$$y(n) = \sum_{k=1}^M a_k y(n-k) + \sum_{k=-N_F}^{N_P} b_k x(n-k)$$

Prendiamone la trasformata  $z$  e riordiniamone le sommatorie:

$$\begin{aligned} Y(z) &= \sum_{n=-\infty}^{\infty} y(n) z^{-n} \\ &= \sum_{n=-\infty}^{\infty} \left[ \sum_{k=1}^M a_k y(n-k) + \sum_{k=-N_F}^{N_P} b_k x(n-k) \right] z^{-n} \\ &= \sum_{k=1}^M a_k \left[ \sum_{n=-\infty}^{\infty} y(n-k) z^{-n} \right] + \sum_{k=-N_F}^{N_P} b_k \left[ \sum_{n=-\infty}^{\infty} x(n-k) z^{-n} \right] \end{aligned}$$

Applicando la regola della trasformata  $z$  di una sequenza ritardata, otteniamo

$$Y(z) = \sum_{k=1}^M a_k z^{-k} Y(z) + \sum_{k=-N_F}^{N_P} b_k z^{-k} X(z)$$

Separiamo gli addendi contenenti  $X(z)$  e  $Y(z)$  e applichiamo la definizione della funzione di trasferimento  $H(z)$ : otteniamo il risultato atteso

$$H(z) = \frac{Y(z)}{X(z)} = \frac{\sum_{k=-N_F}^{N_P} b_k z^{-k}}{1 - \sum_{k=1}^M a_k z^{-k}}$$

**Esempio di calcolo**

Consideriamo l'equazione alle differenze

$$y(n) = a_1 y(n-1) + a_2 y(n-2) + b_0 x(n) + b_1 x(n-1)$$

Prendiamone la trasformata  $z$

$$\begin{aligned} Y(z) &= a_1 z^{-1} Y(z) + a_2 z^{-2} Y(z) + b_0 X(z) + b_1 z^{-1} X(z) \\ &= \frac{(b_0 + b_1 z^{-1})}{(1 - a_1 z^{-1} - a_2 z^{-2})} X(z) \end{aligned}$$

La funzione di trasferimento è quindi

$$H(z) = \frac{b_0 + b_1 z^{-1}}{1 - a_1 z^{-1} - a_2 z^{-2}}$$

## 2.2 Disposizione di poli e zeri in un filtro generico

Ricordiamo la scrittura della funzione di trasferimento come rapporto di due polinomi:

$$X(z) = \frac{P(z)}{Q(z)}$$

Gli zeri, cioè i valori di  $z$  per cui  $P(z) = 0$ , indicano le frequenze complesse in cui il guadagno del filtro è nullo. I poli, cioè i valori di  $z$  per cui  $Q(z) = 0$ , indicano le frequenze complesse in cui il guadagno del filtro è infinito.

Poli e zeri vicini all'origine non modificano significativamente la risposta in frequenza del filtro, mentre zeri vicini alla circonferenza portano a zero le frequenze corrispondenti.

Se la ROC si estende oltre il polo più distante dal centro, allora il sistema è causale. Se la ROC include la circonferenza unitaria, allora il sistema è stabile. Ne segue che un filtro è causale e stabile se i poli sono tutti contenuti nel cerchio unitario.

Poiché ogni filtro può essere scomposto in filtri elementari di primo e secondo ordine, a loro volta distinguibili in ricorsivi e non ricorsivi, sono quattro le combinazioni base di poli e zeri:

- uno zero sull'asse reale in  $z = r_0$  con  $0 < r_0 < \infty$  e il polo corrispondente in  $z = 0$ . La funzione di trasferimento del filtro non ricorsivo di prim'ordine che genera questa disposizione è

$$H_1(z) = 1 - r_0 z^{-1}$$

- un polo sull'asse reale in posizione  $z = r_0$  con  $0 < r_0 < 1$  e il zero corrispondente in  $z = 0$ . La funzione di trasferimento del filtro ricorsivo di prim'ordine che genera questa disposizione è

$$H_2(z) = [1 - r_0 z^{-1}]^{-1}$$

- una coppia coniugata di zeri complessi in  $z = r_0 e^{\pm i\omega_0}$  con  $0 < r_0 < \infty$  e due poli corrispondenti in  $z = 0$ . La funzione di trasferimento del filtro non ricorsivo di second'ordine che genera questa disposizione è

$$H_3(z) = (1 - r_0 e^{i\omega_0} z^{-1})(1 - r_0 e^{-i\omega_0} z^{-1}) = 1 - 2r_0 \cos(\omega_0) z^{-1} + r_0^2 z^{-2}$$

- una coppia coniugata di poli complessi in  $z = r_0 e^{\pm i\omega_0}$  con  $0 < r_0 < 1$  e due zeri corrispondenti in  $z = 0$ . La funzione di trasferimento del filtro ricorsivo di second'ordine che genera questa disposizione è

$$\begin{aligned} H_4(z) &= [(1 - r_0 e^{i\omega_0} z^{-1})(1 - r_0 e^{-i\omega_0} z^{-1})]^{-1} \\ &= [1 - 2r_0 \cos(\omega_0) z^{-1} + r_0^2 z^{-2}]^{-1} \end{aligned}$$

In una funzione di trasferimento, gli zeri e i poli sono in numero uguale se vengono considerati anche i punti di singolarità in  $z = \infty$ . Se, dopo aver analizzato i punti di singolarità finiti, resta un disavanzo di  $n$  zeri, significa che ci sono  $n$  poli in  $z = \infty$ . La funzione di trasferimento di questa componente è

$$H(z) = z^n$$

ed è caratteristica dei sistemi non causali: la risposta all'impulso è anticipata di  $n$  campioni. In caso di disavanzo di  $n$  poli, si avranno  $n$  zeri in  $z = \infty$ : la risposta all'impulso è ritardata di  $n$  campioni.

La funzione di trasferimento complessiva di un filtro costituito da queste componenti è proporzionale al prodotto delle singole funzioni di trasferimento.

### 2.2.1 Grafico della risposta in frequenza e in fase

Mostriamo come poter ottenere con Scilab la risposta in frequenza e in fase di un filtro con funzione di trasferimento  $H(z) = \frac{z}{z-a}$ :

```
// zero della funzione di trasferimento del filtro
a=.5;
// creazione del numeratore
Ncoeff=[0,1];
N=poly(Ncoeff,'z','coeff');
// creazione del denominatore
Dcoeff=[-(a),1];
D=poly(Dcoeff,'z','coeff');
// definizione del range di frequenza
f=(0:.01:.5);
// calcolo della risposta in frequenza
hf=freq(N,D,exp(2*pi*i*f));
// settiamo la finestra per il grafico e lo visualizziamo
xbasc(); xset("font size",4);
xsetech([0,0,1,.5]); plot2d(f,(abs(hf)))
xtitle("Risposta in frequenza")
xsetech([0,.5,1,.5]); plot2d(f,atan(imag(hf),real(hf)));
xtitle("Fase del filtro")
```

Come risultato otteniamo il grafico rappresentato in figura 2.1.

Al fine di evidenziare l'importanza del posizionamento degli zeri nel piano complesso  $z$ , riportiamo i grafici del modulo (figura 2.2) e della fase (figura 2.3) del filtro avente  $H(z) = \frac{z}{z-a}$  per valori variabili dello zero  $a$ .

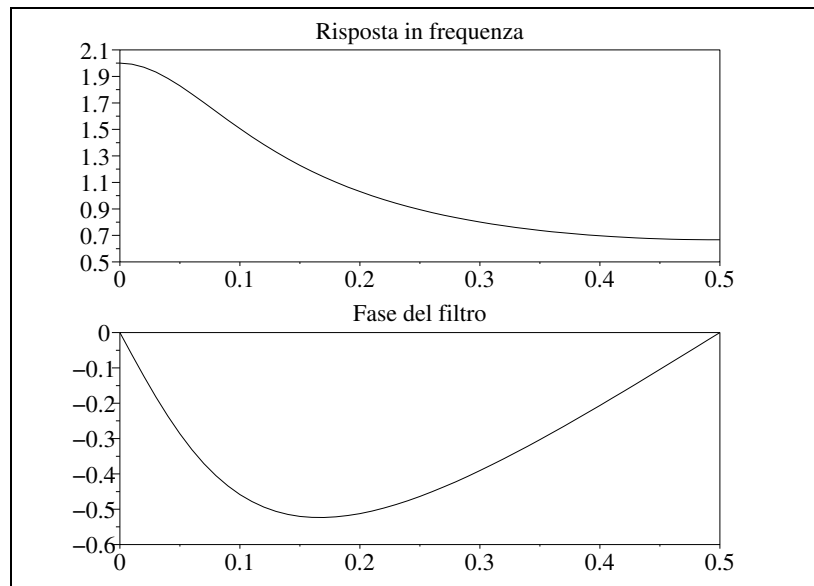


Figura 2.1: Risposta in frequenza e in fase di  $H(z) = \frac{z}{z-0,5}$

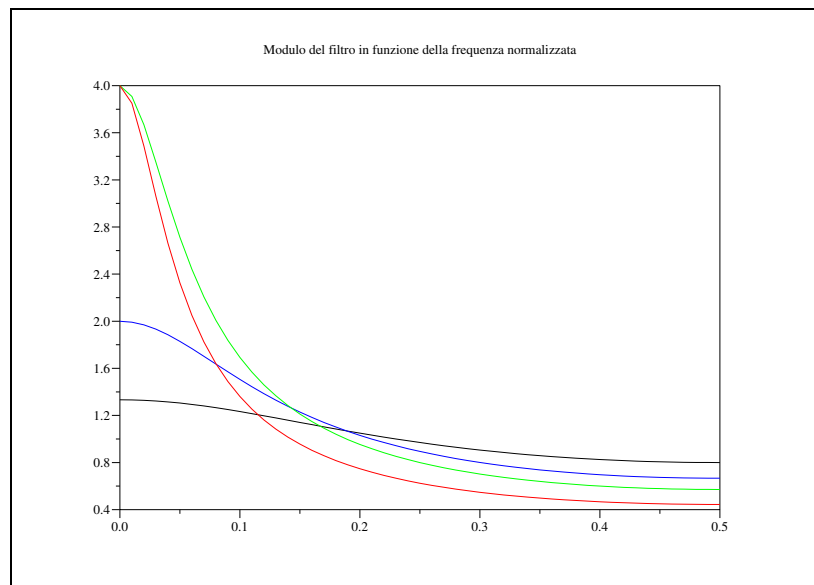


Figura 2.2: Variazione della risposta in frequenza rispetto al valore dello zero (rosso=1.25; verde=0.75; blu=0.5; nero=0.25)

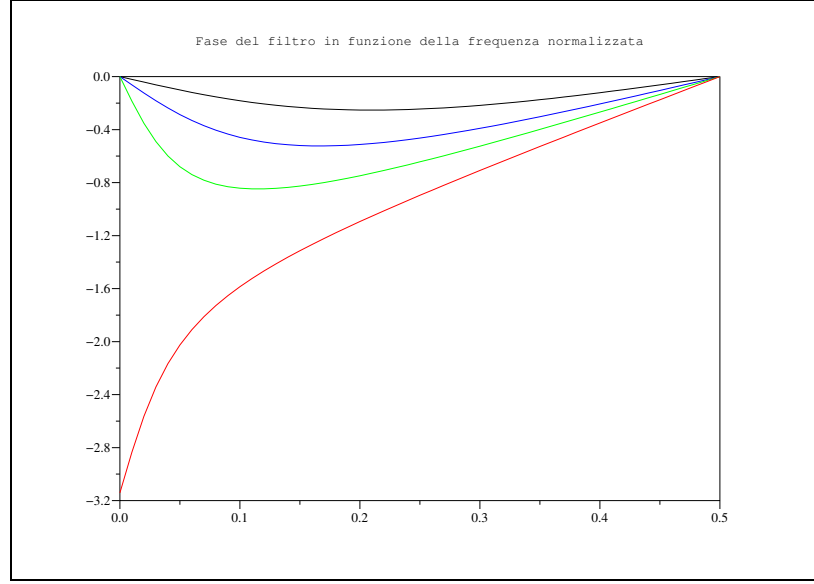


Figura 2.3: Variazione della fase rispetto al valore dello zero (rosso=1.25; verde=0.75; blu=0.5; nero=0.25)

## Disposizione di zeri e poli in un filtro a fase lineare

Gli zeri di una funzione di trasferimento di un filtro lineare seguono una configurazione precisa. Ricordando che un filtro è lineare se è simmetrico, scriviamo la stessa condizione nel dominio  $Z$  e applichiamo la trasformata  $z$ :

$$\begin{aligned} h(n) &= h(N - (1 - n)) \\ H(z) &= z^{-(N-1)} H\left(\frac{1}{z}\right) \end{aligned}$$

Ricordiamo che stiamo considerando un segnale  $h(n)$  a valori reali: quindi, se  $z_0$  è uno zero di  $H(z)$ , allora lo è anche il suo complesso coniugato  $\bar{z}_0$ ; sfruttando la condizione di simmetria precedente, anche  $\left(\frac{1}{z_0}\right)$  e  $\left(\frac{1}{\bar{z}_0}\right)$  sono radici, quindi gli zeri generici di un filtro lineare esistono a gruppi di quattro.

## 2.3 Progetto di filtri mediante disposizione iterativa di poli/zeri

Lo scopo di questa sezione è di determinare la struttura di un filtro e dei suoi coefficienti posizionando in maniera iterativa poli e zeri nel piano  $z$ , osservando il risultato ottenuto dopo il loro posizionamento fino a quando non si realizza il filtro con le specifiche richieste.



Per semplificare la realizzazione del filtro si possono tener presente le seguenti regole e consigli:

#### *Regole*

- Le specifiche della risposta in frequenza consistono in regioni passa-banda e taglia-banda (vedi figura 2.4). Queste bande sono mappate sulla circonferenza di raggio unitario con lo scopo di facilitare il posizionamento dei poli e degli zeri.
- Tutti i poli dovranno avere un raggio  $r_p$  compreso nell'intervallo  $0,6 \leq r_p < 0,96$ .  
Il limite superiore garantisce che nella fase di realizzazione pratica eventuali errori di troncamento o arrotondamento causino l'instabilità del filtro mentre il limite inferiore serve ad evitare di avere poli che diano una scarsa risposta in frequenza.
- La progettazione del filtro viene ottimizzata analizzando, dopo ogni posizionamento di poli e zeri, il grafico della risposta in frequenza. Operando in questo modo si può rimuovere la singolarità o modificarne la posizione nel caso in cui non si ottenga l'effetto desiderato.

#### *Consigli*

- Lo sviluppo dovrebbe iniziare posizionando un polo a distanza  $r_p$  in mezzo ad ogni zona passa-banda e uno zero in mezzo ad ogni zona taglia-banda sulla circonferenza unitaria.
- La risposta in frequenza al di sopra di una limitata banda di frequenze può essere incrementata inserendo, nella circonferenza di raggio unitario, un polo nell'angolo corrispondente al centro della banda interessata. La dimensione del raggio del polo determinerà l'aumento sia della banda sia del guadagno.
- La risposta in frequenza al di sopra di una banda limitata può essere diminuita posizionando uno zero nell'angolo corrispondente al centro della banda. Per ottenere un'attenuazione massima, lo zero dovrà essere posizionato sulla circonferenza di raggio unitario.

### 2.3.1 Realizzazione iterativa di un filtro passa-basso

Proveremo ora a realizzare un filtro seguendo le indicazioni precedentemente citate. Le caratteristiche del filtro passa-basso sono:

passa-banda

$$-1 < |H(e^{j\omega})|_{dB} \leq 0 \quad \text{per } 0 \leq \omega \leq \frac{\pi}{4}$$

taglia-banda

$$|H(e^{j\omega})|_{dB} \leq -50 \quad \text{per } \frac{\pi}{2} \leq \omega \leq \pi$$

Le zone di interesse sono evidenziate nella figura 2.4.

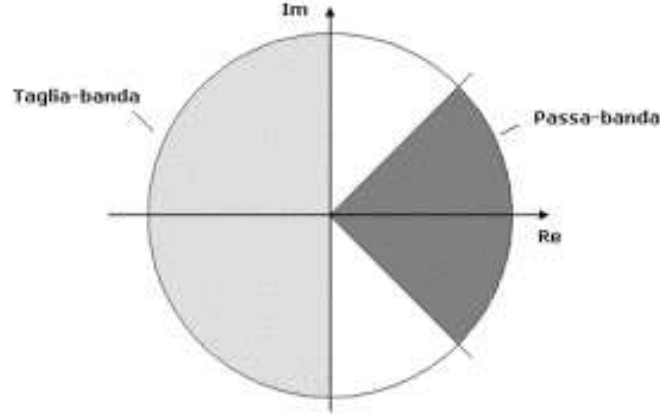


Figura 2.4: Settori in cui verranno disposti i poli e gli zeri

### Passi della realizzazione

- *Singolarità 1 e 2*

Posizionare un polo con raggio  $r = 0.7$  in mezzo alla zona passa-banda e uno zero in mezzo alla zona taglia-banda, sulla circonferenza unitaria. La funzione di trasferimento del filtro risulta

$$H(z) = \frac{1 + z^{-1}}{1 - 0.7z^{-1}}$$

Il grafico della risposta in frequenza del filtro è rappresentato in figura 2.5.

- *Singolarità 3*

Inserire uno zero in testa alla zona taglia-banda, sulla circonferenza unitaria.

La funzione di trasferimento del filtro risulta

$$H(z) = \frac{(1 + z^{-1})(1 + z^{-2})}{1 - 0.7z^{-1}}$$

Il grafico della risposta in frequenza del filtro è rappresentato in figura 2.6.

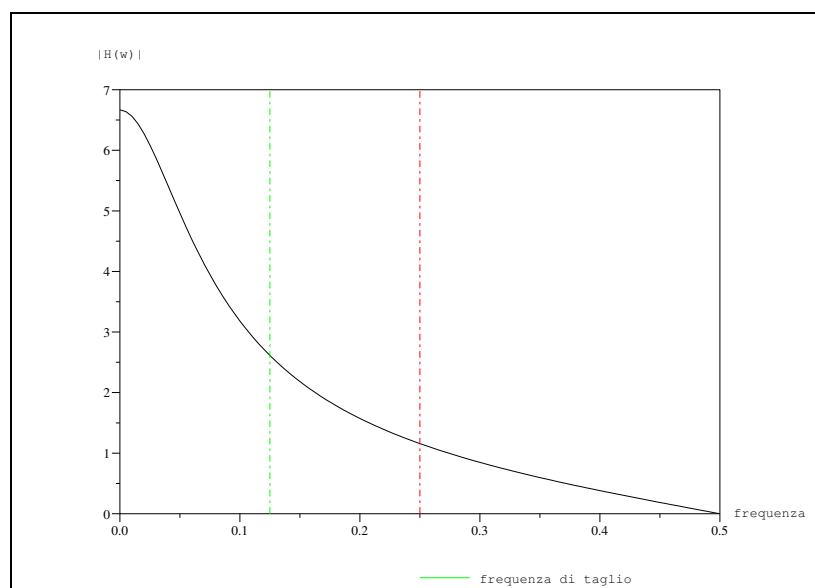


Figura 2.5: Risposta in frequenza del filtro dopo il posizionamento delle singolarità 1 e 2

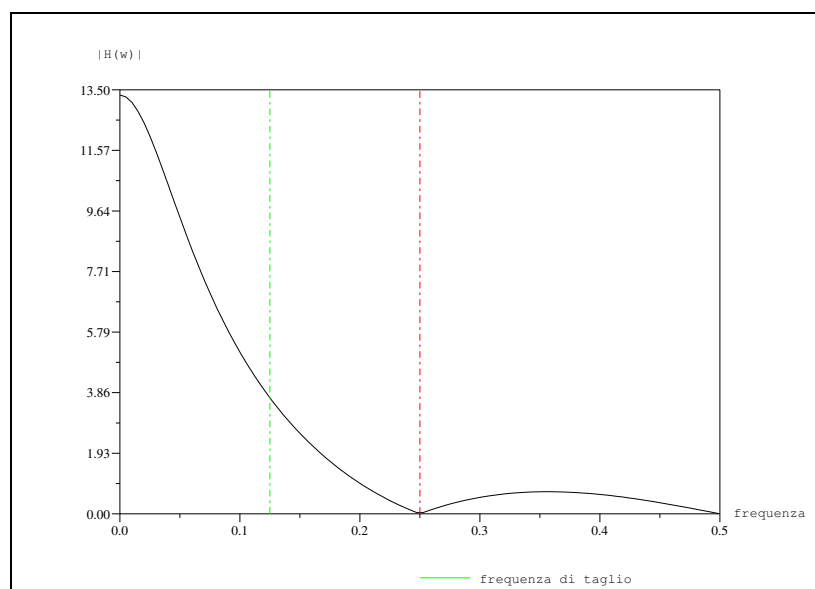


Figura 2.6: Risposta in frequenza del filtro dopo il posizionamento della singolarità 3

- *Singolarità 4*

Aggiungere un polo a  $z = 0,9e^{\pm j50^\circ}$  per sollevare la risposta in frequenza nella zona passa-banda.

La funzione di trasferimento del filtro risulta

$$H(z) = \frac{(1 + z^{-1})(1 + z^{-2})}{(1 - 0,7z^{-1})(1 - 1,16z^{-1} + 0,81z^{-2})}$$

Il grafico della risposta in frequenza del filtro è rappresentato in figura 2.7.

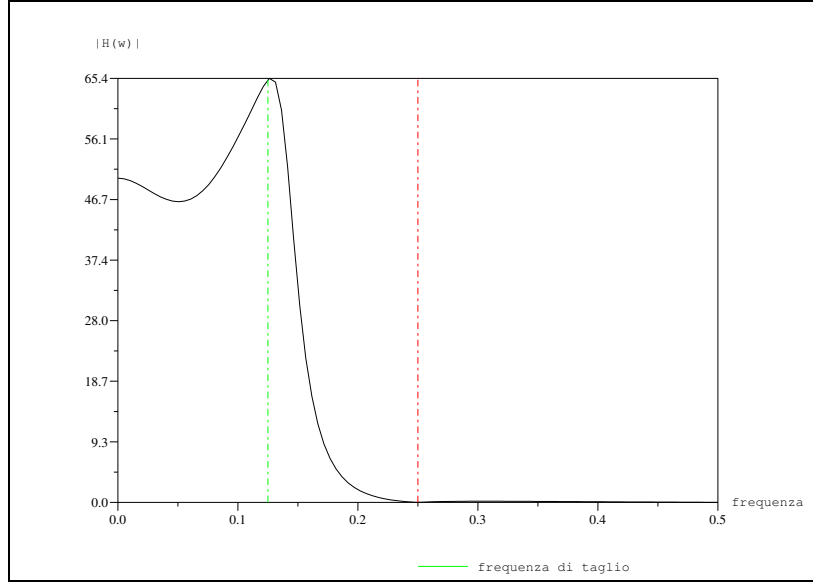


Figura 2.7: Risposta in frequenza del filtro dopo il posizionamento della singolarità 4

- *Singolarità 5*

Posizionare uno zero sulla circonferenza unitaria, dove risulta massima la risposta nella zona taglia-banda. Questo zero limita l'incremento dovuto alla singolarità 4.

La funzione di trasferimento del filtro risulta

$$H(z) = \frac{(1 + z^{-1})(1 + z^{-2})(1 + 1,03z^{-1} + z^{-2})}{(1 - 0,7z^{-1})(1 - 1,16z^{-1} + 0,81z^{-2})}$$

Il grafico della risposta in frequenza del filtro è rappresentato in figura 2.8.

- *Singolarità 6*

Aggiungere un polo in  $z = 0,75e^{\pm j40^\circ}$  per incrementare la risposta in

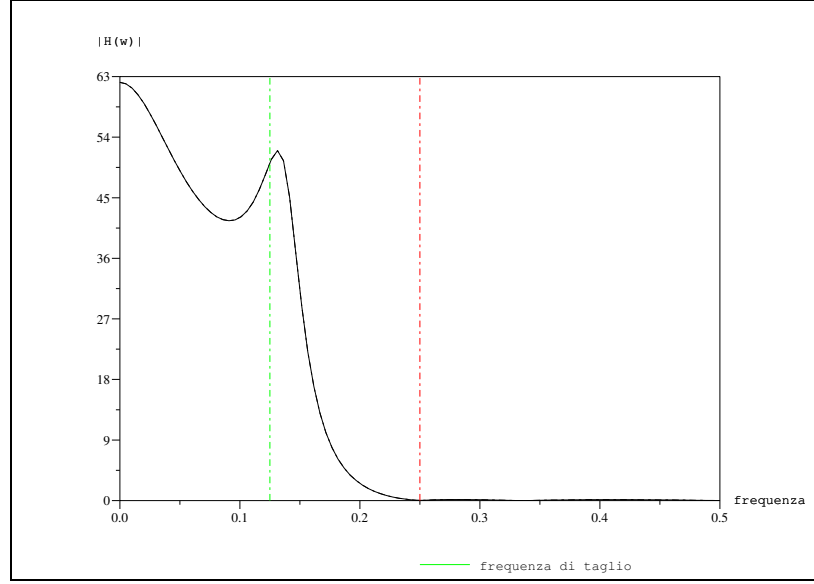


Figura 2.8: Risposta in frequenza del filtro dopo il posizionamento della singolarità 5

frequenza in testa alla zona passa-banda.

La funzione di trasferimento del filtro risulta

$$H(z) = \frac{(1 + z^{-1})(1 + z^{-2})(1 + 1,03z^{-1} + z^{-2})}{(1 - 0,7z^{-1})(1 - 1,16z^{-1} + 0,81z^{-2})(1 - 1,15z^{-1} + 0,56z^{-2})}$$

Il grafico della risposta in frequenza del filtro è rappresentato in figura 2.9.

- *Singolarità 7*

Posizionare un polo in  $z = 0,6e^{\pm j38^\circ}$  per incrementare la risposta in frequenza in mezzo alla zona passa-banda.

La funzione di trasferimento finale del filtro risulta

$$H(z) = \frac{(1+z^{-1})(1+z^{-2})(1+1,03z^{-1}+z^{-2})}{(1-0,7z^{-1})(1-1,16z^{-1}+0,81z^{-2})(1-1,15z^{-1}+0,56z^{-2})(1-0,95z^{-1}+0,36z^{-2})}$$

Alla funzione di trasferimento del filtro bisogna aggiungere un guadagno pari a 0,0027; questo valore permette di normalizzare il guadagno.

### Grafico finale della risposta in frequenza

Per realizzare il grafico con Scilab si possono utilizzare i seguenti comandi

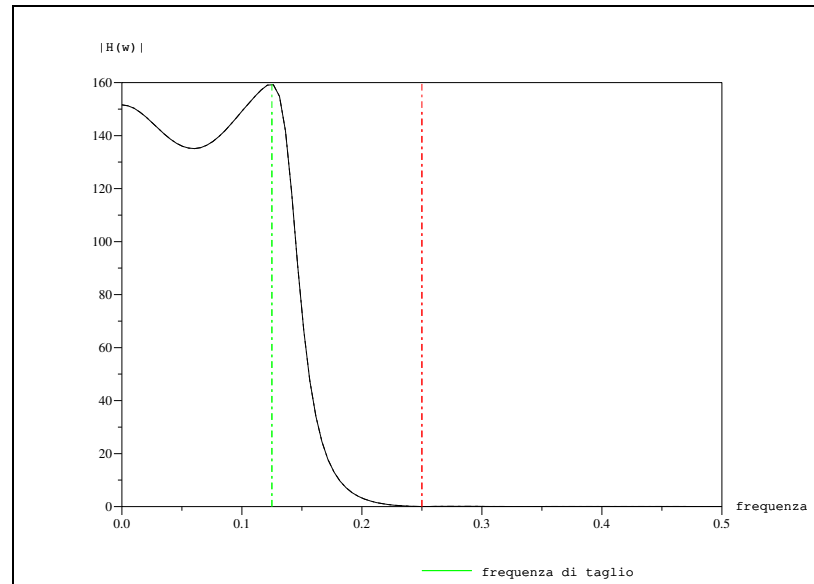


Figura 2.9: Risposta in frequenza del filtro dopo il posizionamento della singolarità 6

```
// definiamo il numeratore della funzione di trasferimento
num_Hz=(1+%z^(-1))*(1+%z^(-2))*(1+1.03*%z^(-1)+%z^(-2));
// fattorizziamo il lungo denominatore in tre termini
den1_Hz=(1-0.7*%z^(-1))*(1-1.16*%z^(-1)+0.81*%z^(-2));
den2_Hz=(1-1.15*%z^(-1)+0.56*%z^(-2));
den3_Hz=(1-0.95*%z^(-1)+0.36*%z^(-2));
// componiamo il denominatore
den_Hz=den1_Hz*den2_Hz*den3_Hz;
// infine creiamo la funzione di trasferimento
Hz=num_Hz/den_Hz;
A=0.0027;
// calcoliamo la risposta in frequenza
[xm,fr]=frmag(A*Hz,100);
// Disegna il grafico del modulo della risposta in frequenza
plot2d(fr,xm);
xtitle(xtit="",xax="frequenza",yax="|H(w)|");
xset("line style",3);
y=(0:100);
xt=(.125*ones(y));
// Disegna la frequenza di taglio
plot2d(xt,y,rect=[0,0,.5,1],style=(3),leg="frequenza taglio");
xs=(.25*ones(y));
```

```
// Disegna la frequenza di stop
plot2d(xs,y,rect=[0,0,.5,1],style=(5));
```

Il grafico finale della risposta in frequenza del filtro è rappresentato in figura 2.10.

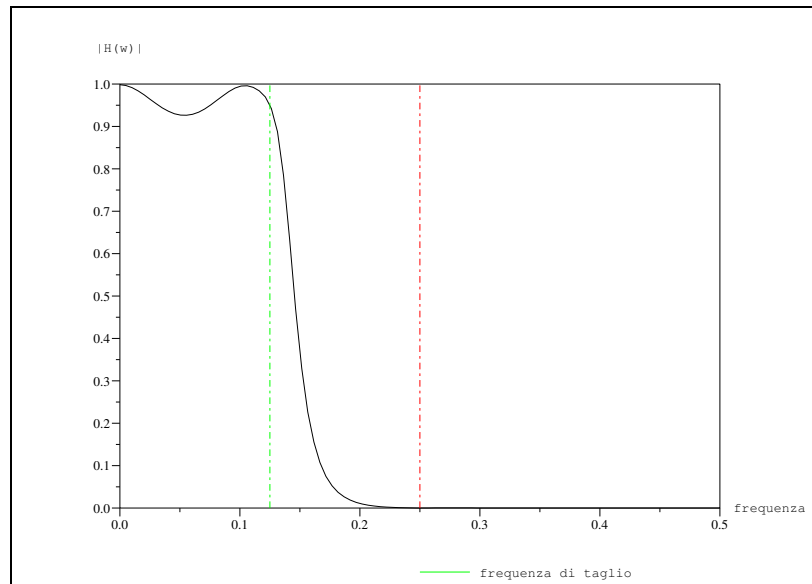


Figura 2.10: Risposta in frequenza finale del filtro.

### Grafico finale dei poli e degli zeri

Per realizzare il grafico con Scilab si possono utilizzare i seguenti comandi

```
// Grafico poli/zeri
num_Hz=(1+%z^(-1))*(1+%z^(-2))*(1+1.03*%z^(-1)+%z^(-2));
// fattorizziamo il lungo denominatore in tre termini
den1_Hz=(1-0.7*%z^(-1))*(1-1.16*%z^(-1)+0.81*%z^(-2));
den2_Hz=(1-1.15*%z^(-1)+0.56*%z^(-2));
den3_Hz=(1-0.95*%z^(-1)+0.36*%z^(-2));
// componiamo il denominatore
den_Hz=den1_Hz*den2_Hz*den3_Hz;
Hz=num_Hz/den_Hz;
xbasc(); plzr(Hz)
```

Il grafico dei poli/zeri del filtro è rappresentato in figura 2.11.

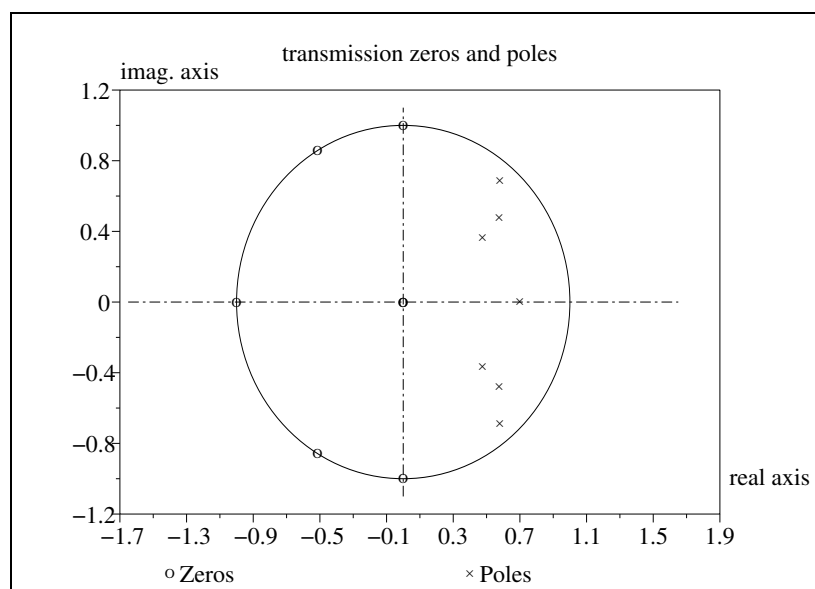


Figura 2.11: Grafico poli/zeri del filtro.



## Capitolo 3

# Disposizione di poli e zeri nei filtri FIR

I filtri FIR sono sistemi LTI aventi risposta finita all'impulso e sono rappresentati da un polinomio in  $z^{-1}$ :

$$H(z) = \sum_{k=0}^{M-1} h(k)z^{-k}$$

Il diagramma poli/zeri corrispondente presenta un polo in  $z = 0$ , quindi interno alla circonferenza di raggio unitario: i filtri FIR sono sempre causali e stabili, inoltre hanno fase lineare se la funzione di risposta all'impulso è simmetrica o antisimmetrica.

Vediamone qualche esempio, usando Scilab come strumento di analisi.

### 3.1 Filtro con uno zero sulla circonferenza unitaria

Verifichiamo innanzitutto che uno zero in  $z = e^{i\omega_0}$  modifica la risposta in frequenza azzerandola nella frequenza corrispondente a  $\omega_0$ . Un filtro di questo tipo ha la funzione di trasferimento  $H(z) = \frac{1-z^{-1}}{z-1}$  e il diagramma poli/zeri può essere ottenuto in Scilab con i seguenti comandi:

```
// inizializziamo il sistema
clear;
// creiamo la funzione di trasferimento:
// definiamo il vettore di coefficienti del numeratore
numcoeff=[1 -1];
// creiamo il polinomio al numeratore
num=poly(numcoeff,'z','c');
// definiamo il vettore di coefficienti del denominatore
dencoeff=[0 1];
// creiamo il polinomio al denominatore
```

```

den=poly(dencoeff,'z','c');
// creiamo la funzione di trasferimento
combsl=syslin('d',num,den);
// inizializziamo la finestra grafica
xbasc();
// visualizziamo il grafico poli/zeri
plzr(combsl);

```

Otteniamo il risultato di figura 3.1.

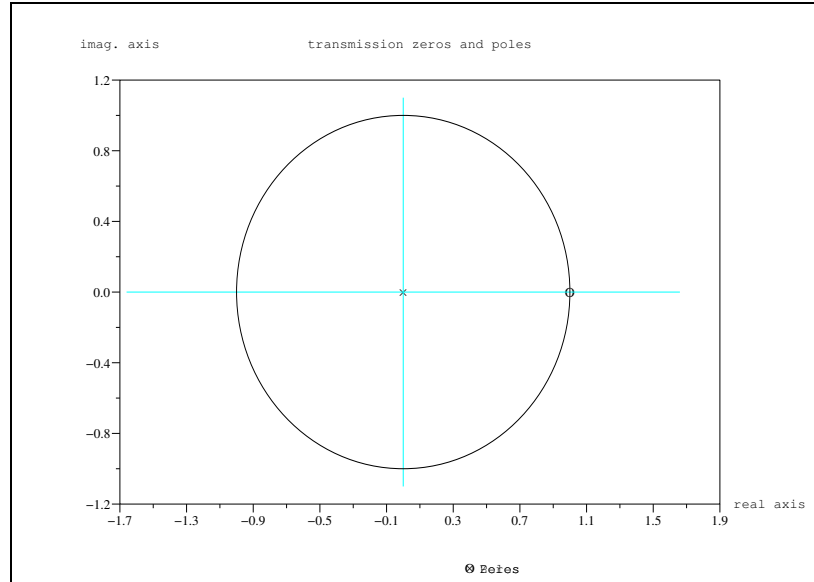


Figura 3.1: Grafico poli/zeri

Calcoliamo ora la risposta in frequenza:

```

// definiamo il range discreto delle frequenze
f=(-.5:.001:.5);
// calcoliamo la risposta in frequenza
hf=freq(num,den,exp(2*pi*i*f));
// visualizziamo il grafico
xbasc(); plot2d(f,abs(hf))

```

Il filtro ha la risposta in frequenza attesa, come dimostra il grafico 3.2.

## 3.2 Filtri COMB

Un filtro COMB di ordine  $M$  ha funzione di trasferimento

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

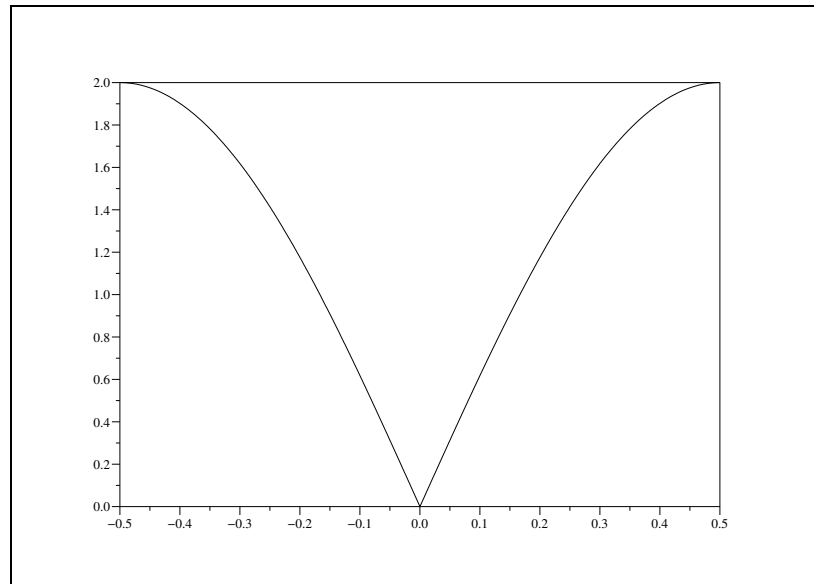


Figura 3.2: Risposta in frequenza

In Scilab, il suo diagramma poli/zeri può essere ottenuto con i seguenti comandi:

```
// inizializziamo il sistema
clear;
// creiamo la funzione di trasferimento:
// definiamo il vettore di coefficienti del numeratore
numcoeff=[-1 0 0 0 0 0 0 1];
// creiamo il polinomio al numeratore
num=poly(numcoeff,'z','c');
// definiamo il vettore di coefficienti del denominatore
dencoeff=[0 0 0 0 0 0 0 8];
// creiamo il polinomio al denominatore
den=poly(dencoeff,'z','c');
// creiamo la funzione di trasferimento come sistema lineare
combsl=syslin('d',num,den);
// inizializziamo la finestra grafica
xbasc();
// visualizziamo il grafico poli/zeri
plzr(combsl);
```

Otteniamo il risultato rappresentando nella figura 3.3.

La risposta in frequenza può essere determinata con i seguenti comandi:

```
// definiamo il range discreto delle frequenze
```

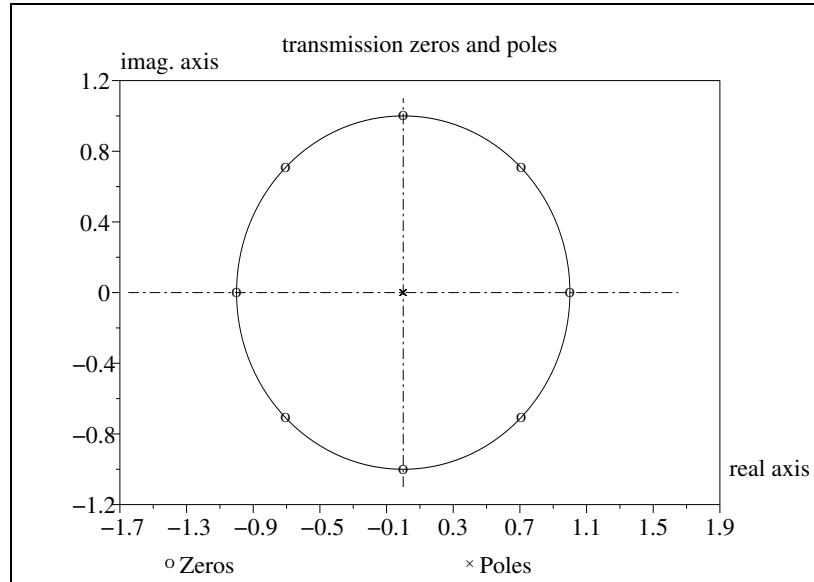


Figura 3.3: Grafico poli/zeri di un filtro COMB di ordine 8

```
f=(-.5:.001:.5);
// calcoliamo la risposta in frequenza
hf=freq(num,den,exp(2*pi*i*f));
// inizializziamo la finestra grafica e
// visualizziamo la risposta in frequenza
xbasc(); plot2d(f,abs(hf))
```

Il grafico è mostrato in figura 3.4.

### Filtro passa-basso implementato con un filtro COMB

È possibile modificare un filtro COMB in modo da filtrare determinate frequenze ed utilizzarlo quindi come un comune filtro passa-basso o passa-alto di facile implementazione ma scarsa efficacia. Proviamo ad esempio ad eliminare il polo in  $z = 1$ : la funzione di trasferimento del filtro ottenuto sarà

$$H(z) = \frac{1}{8} \frac{z^8 - 1}{z^7 - z^6}$$

Costruiamo il grafico poli/zeri di figura 3.5:

```
clear;
numcoeff=[-1 0 0 0 0 0 0 0 1];
num=poly(numcoeff,'z','c');
dencoeff=[0 0 0 0 0 0 0 -1/8 1/8];
den=poly(dencoeff,'z','c');
```

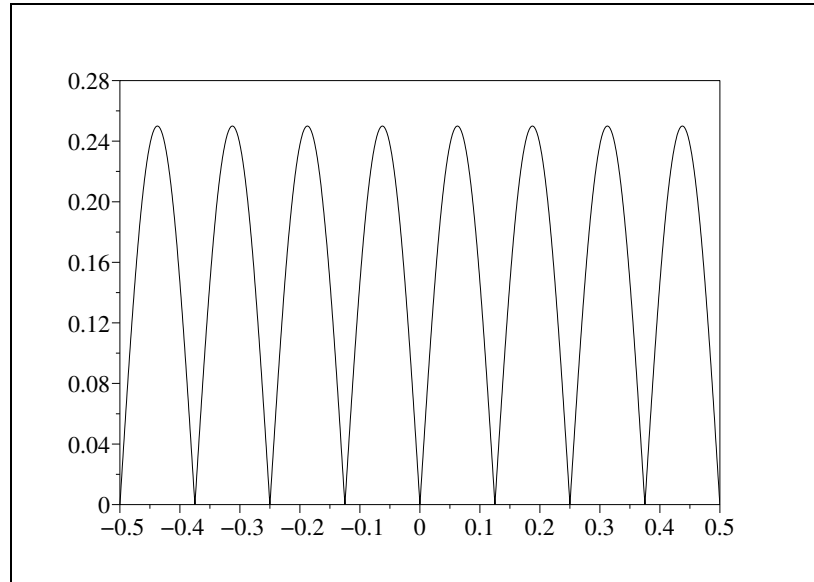


Figura 3.4: Risposta in frequenza di un filtro COMB di ordine 8

```
combsl=syslin('d',num,den);
xbasc();plzr(combsl);
```

La corrispondente risposta in frequenza si ottiene con i comandi

```
f=(-.5:.001:.5);
hf=freq(num,den,exp(2*pi*i*f));
xbasc(); xset("font size",4); plot2d(f,abs(hf))
```

ed è mostrata nella figura 3.6.

### 3.3 Filtri realizzati con il metodo delle finestre

Progettiamo un filtro con il metodo delle finestre: grazie a Scilab, è facile ottenerne i coefficienti usando la funzione `wfir`, che richiede come parametri

- il tipo di filtro: **lp** (passa-basso), **hp** (passa-alto), **bp** (passa-banda), **sb** (taglia-banda)
- l'ordine del filtro
- una o due frequenze di taglio, in base al tipo di filtro scelto
- il tipo di finestra da applicare: **re** (rettangolare), **tr** (triangolare), **hm** (Hamming), **hn** (Hanning), **kr** (Kaiser), **ch** (Chebyshev)
- i parametri della finestra scelta

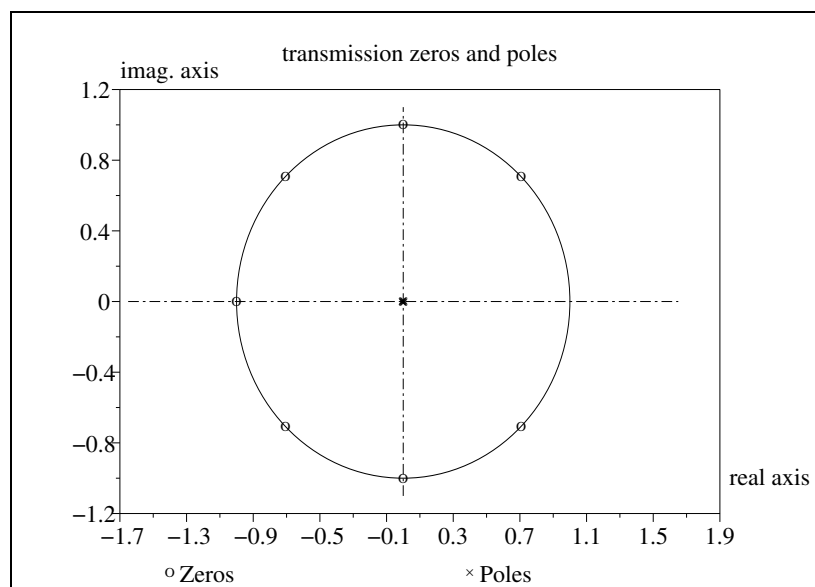


Figura 3.5: Grafico poli/zeri di un filtro passa-basso implementato con un filtro COMB di ordine 8

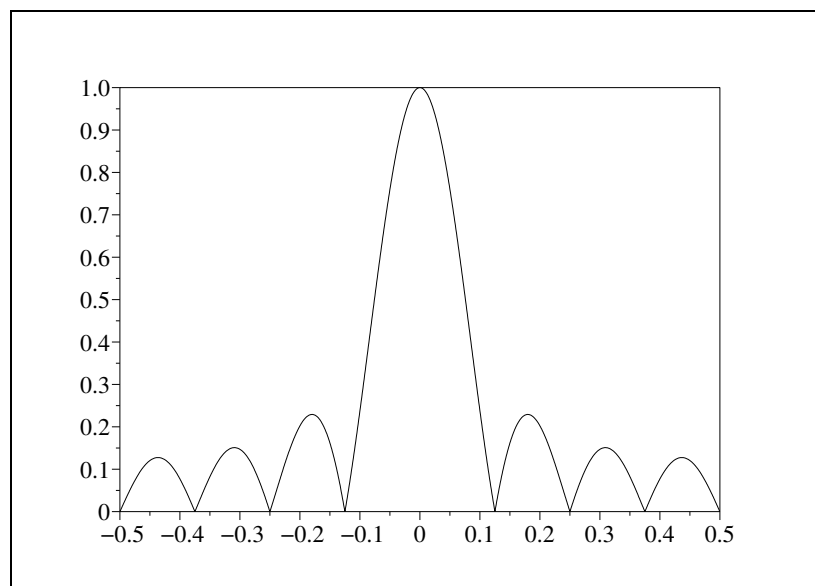


Figura 3.6: Risposta in frequenza di un filtro passa-basso implementato con un filtro COMB di ordine 8

e restituisce in output

- i coefficienti del filtro nel dominio del tempo
- la risposta del filtro nel dominio della frequenza
- i punti del dominio della frequenza nei quali la risposta viene valutata

Vediamone un paio di esempi.

### Filtro passa-basso realizzato con una finestra rettangolare

Creiamo un filtro passa-basso di ordine 20 con frequenza di taglio pari a 0.3 utilizzando una finestra rettangolare:

```
clear;
[valcoeff,filtamp,filtfreq]=wfir('lp',20,[.3 0],'re',[0 0]);
num=poly(valcoeff,'z','c');
xbasc();
// la funzione 'horner' valuta il polinomio num in 1/z
plzr(horner(num,1/%z));
```

Il grafico poli/zeri risultante è mostrato in figura 3.7.

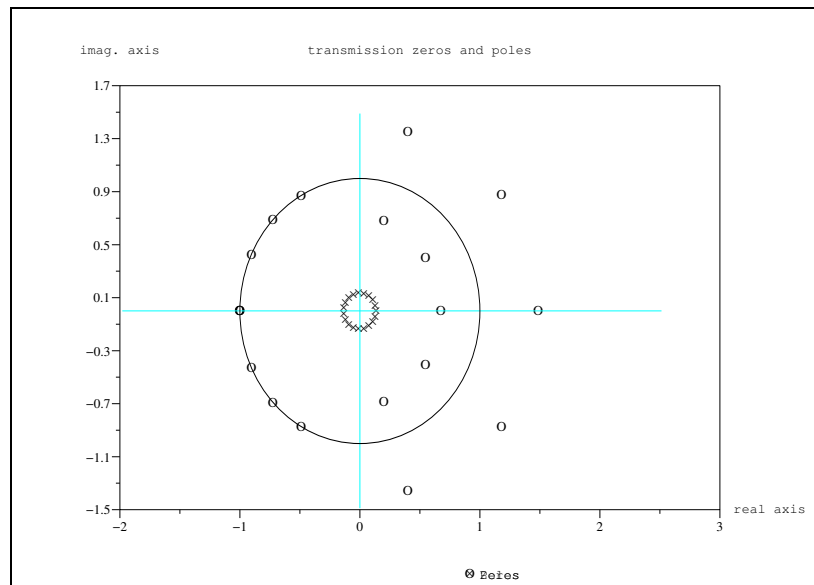


Figura 3.7: Grafico poli/zeri di un filtro passa-basso di ordine 20 realizzato con una finestra rettangolare

La risposta in frequenza (figura 3.8) può essere ottenuta con le seguenti istruzioni:

```
xbasc(); plot2d(filtfreq,filtamp);
```

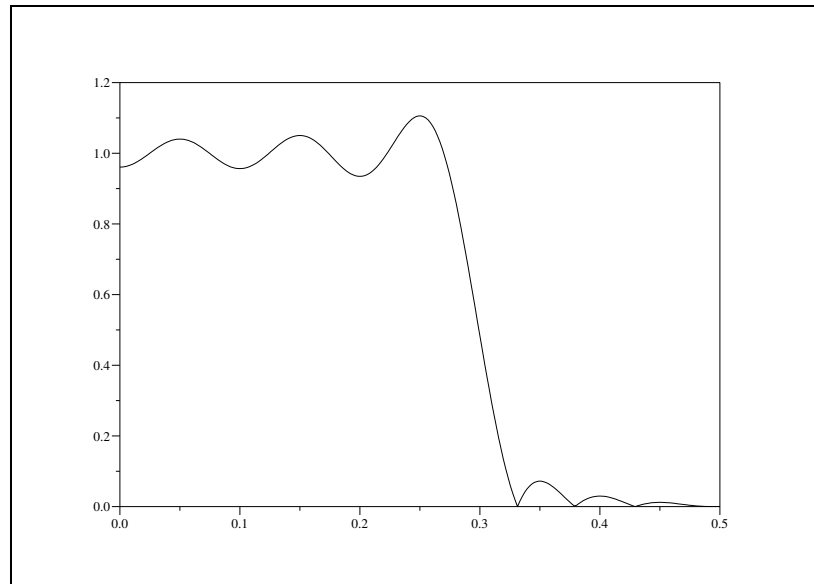


Figura 3.8: Risposta in frequenza di un filtro passa-basso di ordine 20 realizzato con una finestra rettangolare

### Filtro passa-banda realizzato con una finestra di Hamming

Creiamo un filtro passa-banda di ordine 20 con frequenze di taglio pari a 0.2 e 0.37 utilizzando una finestra di Hamming:

```
clear;
[coeff,amp,freq]=wfir('bp',20,[.2 .37],'hm',[0 0]);
num=poly(coeff,'z','c');
xbasc(); plzr(horner(num,1/%z));
```

Il grafico poli/zeri del filtro corrispondente è mostrato in figura 3.9.

La risposta in frequenza (figura 3.10) può essere ottenuta con le seguenti istruzioni:

```
xbasc(); plot2d(freq,amp);
```

## 3.4 Filtro passa-alto realizzato con il metodo minimax

Progettiamo un filtro con il metodo dell'approssimazione minimax: grazie a Scilab, è facile ottenerne i coefficienti usando la funzione `eqfir`, che richiede come parametri

- il numero dei punti di cui sarà composto il filtro risultante



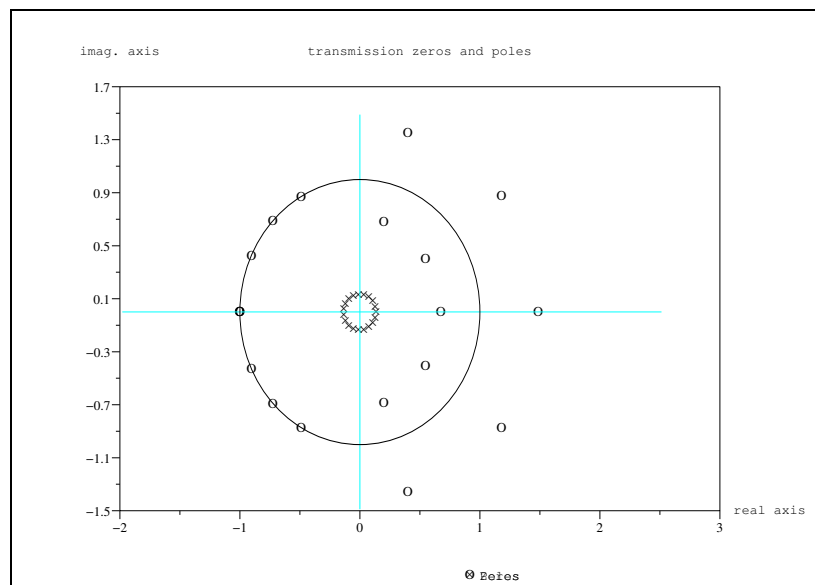


Figura 3.9: Grafico poli/zeri di un filtro passa-banda di ordine 20 realizzato con una finestra di Hamming

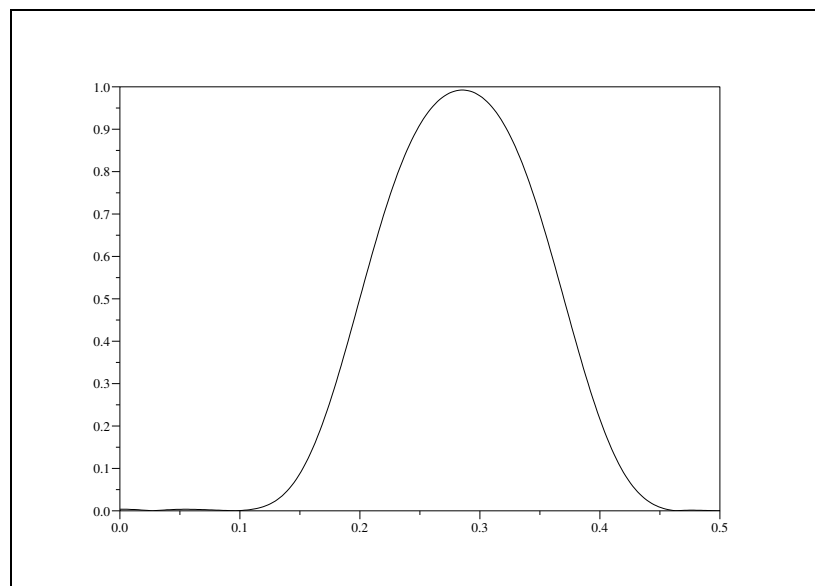


Figura 3.10: Risposta in frequenza di un filtro passa-banda di ordine 20 realizzato con una finestra di Hamming

- $M$  coppie di frequenze delimitanti le  $M$  bande
- $M$  magnitudini, un valore per ciascuna banda
- $M$  valori rappresentanti il peso relativo dell'errore in ciascuna banda

e restituisce in output i coefficienti del filtro.

Vediamone un esempio: progettiamo un filtro di 33 elementi, con banda azzerata nell'intervallo  $(0, .22)$  e passante nell'intervallo  $(.25, .5)$ .

```
clear;
// cerchiamo i coefficienti del filtro richiesto
hn=eqfir(33,[0 .22;.24 .5],[0 1],[.1 1]);
// creiamo ora la funzione di trasferimento
filter=poly(hn,'z','c');
```

Ottenuta la funzione di trasferimento, procediamo nel consueto modo per ottenere il grafico poli/zeri:

```
xbasc(); plzr(horner(filter,1/%z));
```

Il risultato è visualizzato in figura 3.11.

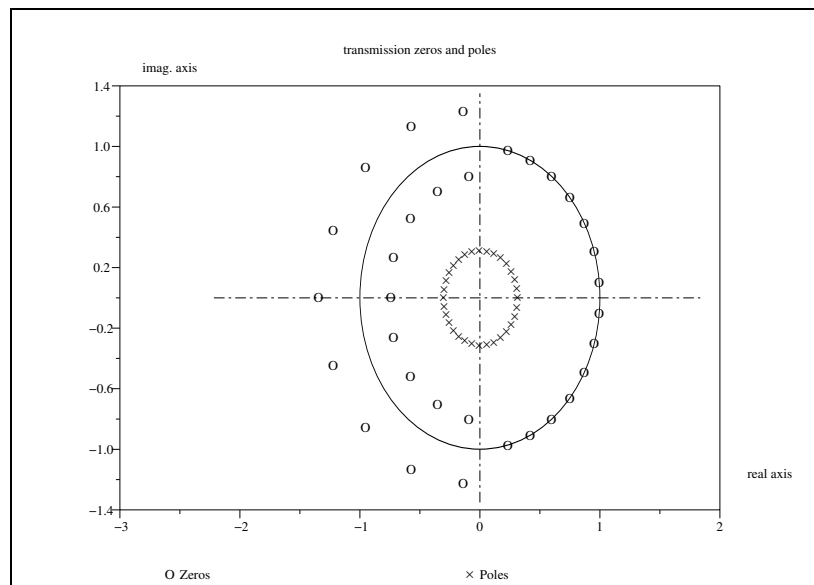


Figura 3.11: Grafico poli/zeri di un filtro passa-alto realizzato con il metodo di approssimazione minimax

La risposta in frequenza è calcolato con le seguenti righe di codice:

```
[hm,fr]=frmag(hn,256);
xbasc(); plot2d(fr,hm);
```

Il suo grafico è in figura 3.12.

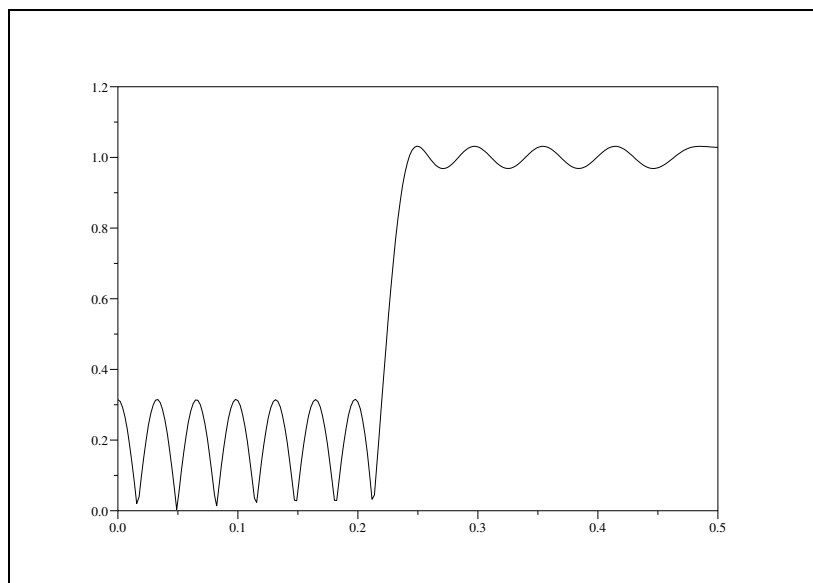


Figura 3.12: Risposta in frequenza di un filtro passa-alto realizzato con il metodo di approssimazione minimax

## Capitolo 4

# Disposizione di poli e zeri nei filtri IIR

I filtri IIR (Infinitive Impulse Response) sono filtri appartenenti alla sotto-classe dei sistemi LTI causali caratterizzati da una risposta infinita all'impulso.

Questi filtri vengono rappresentati nel dominio delle frequenze con una funzione di trasferimento razionale in  $z^{-1}$  della forma

$$H(z) = \frac{\sum_{k=0}^{M-1} b_k z^{-k}}{\sum_{k=0}^{L-1} a_k z^{-k}}$$

Utilizzando Scilab, è possibile progettare un filtro IIR ed ottenerne i coefficienti grazie alla funzione `iir`, dove

- l'ordine del filtro
- il tipo di filtro: `lp` (passa-basso), `hp` (passa-alto), `bp` (passa-banda), `sb` (taglia-banda)
- il corrispondente design analogico: `butt` (Butterworth), `cheb1` (Chebyshev, oscillazioni solo in banda passante), `cheb2` (Chebyshev, oscillazioni solo in banda proibita), `ellip` (ellittico)
- una o due frequenze di taglio, in base al tipo di filtro scelto
- gli errori ammessi nelle bande definite

### 4.1 Filtri di Butterworth

I filtri di Butterworth sono caratterizzati da un discreto guadagno in banda passante, da un sufficiente guadagno in banda di transizione e da una fase

non lineare ma comunque di discreta qualità.

La forma generale del modulo del filtro di Butterworth è del tipo

$$\frac{1}{\left|B_N\left(\frac{\omega}{\omega_c}\right)\right|} = \frac{1}{\sqrt{1 + \left(\frac{\omega}{\omega_c}\right)^{2N}}}$$

dove  $N$  è l'ordine del filtro,  $\omega_c$  è la frequenza di taglio e  $B_N$  è il polinomio  $N - esimo$  di Butterworth.

Creiamo un filtro Butterworth passa-basso di ordine 3 con frequenza di taglio 0,15 e frequenza di stop 0,3.

Il diagramma dei poli/zeri e della risposta in frequenza può essere ottenuto in Scilab con i seguenti comandi

```
// filtro BUTTERWORTH
[hz]=iir(3,'lp','butt',[0.125 .3],[0.2 0.2]);
[hzm,fr]=frmag(hz,100);
xbasc();xset('window',0);
// grafico risposta in frequenza
plot2d(fr,hzm,rect=[0,0,.5,1.1]);
xtitle(tit="",xax="frequenza",yax="|H(w)|")
// grafico poli/zeri
xset('window', 1);
plzr(hz);
```

Otteniamo così il risultato di figura 4.2 e figura 4.1.

## 4.2 Filtri di Chebyshev

La caratteristica principale dei filtri di Chebyshev è quella di distribuire l'accuratezza dell'approssimazione uniformemente lungo tutta la banda passante (tipo I) o quella proibita (tipo II) scegliendo un'approssimazione con oscillazioni della stessa ampiezza su tutta la banda.

La forma generale del modulo del filtro di Chebyshev è del tipo

$$\frac{1}{\sqrt{1 + \epsilon^2 C_N^2\left(\frac{\omega}{\omega_c}\right)}}$$

dove  $N$  è l'ordine del filtro,  $\omega_c$  è la frequenza di taglio,  $\epsilon$  è una costante minore di 1 e  $C_N$  è il polinomio  $N - esimo$  di Chebyshev.

### 4.2.1 Filtro di Chebyshev - tipo I

Creiamo un filtro di Chebyshev taglia-banda di ordine 10 con frequenza di taglio di 0,2 e 0,35.

Il diagramma dei poli/zeri e della risposta in frequenza può essere ottenuto in Scilab con i seguenti comandi

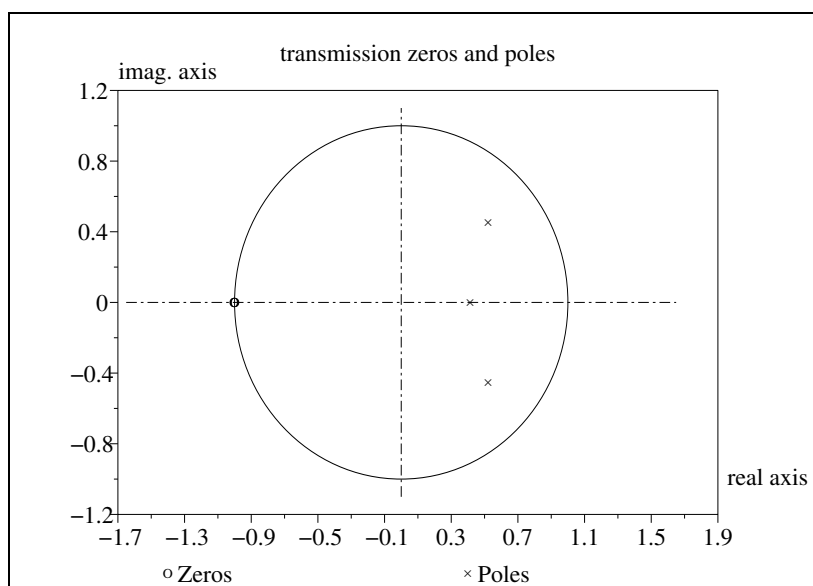


Figura 4.1: Grafico poli/zeri del filtro di Butterworth

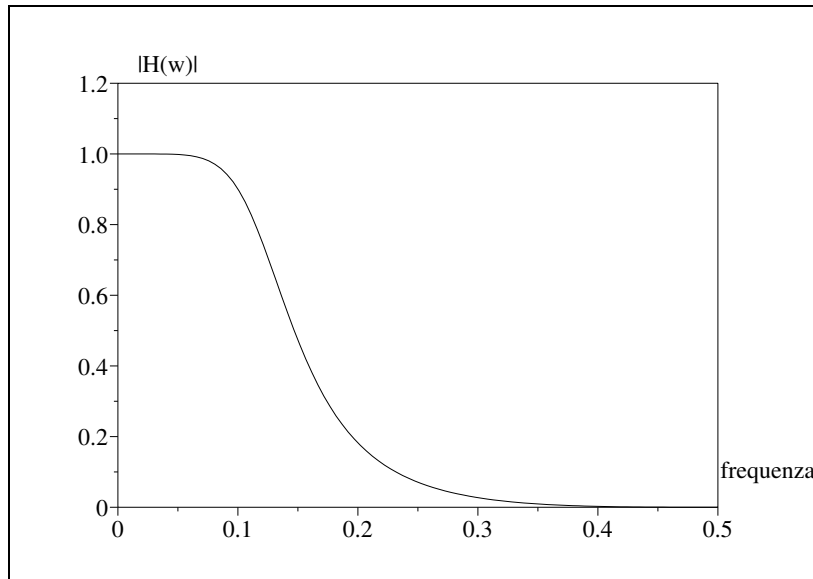


Figura 4.2: Risposta in frequenza del filtro di Butterworth

```
// filtro CHEBYTCHEV tipo I
[hz]=iir(10,'sb','cheb1',[.2 .35],[0.1 0.1]);
[hzm,fr]=frmag(hz,100);
xbasc();xset('window',0);
// grafico risposta in frequenza
plot2d(fr,hzm,rect=[0,0,.5,1.1]);
xtitle(tit="",xax="frequenza",yax="|H(w)|")
// grafico poli/zeri
xset('window', 1);
plzr(hz);
```

Otteniamo così il risultato di figura 4.4 e figura 4.3.

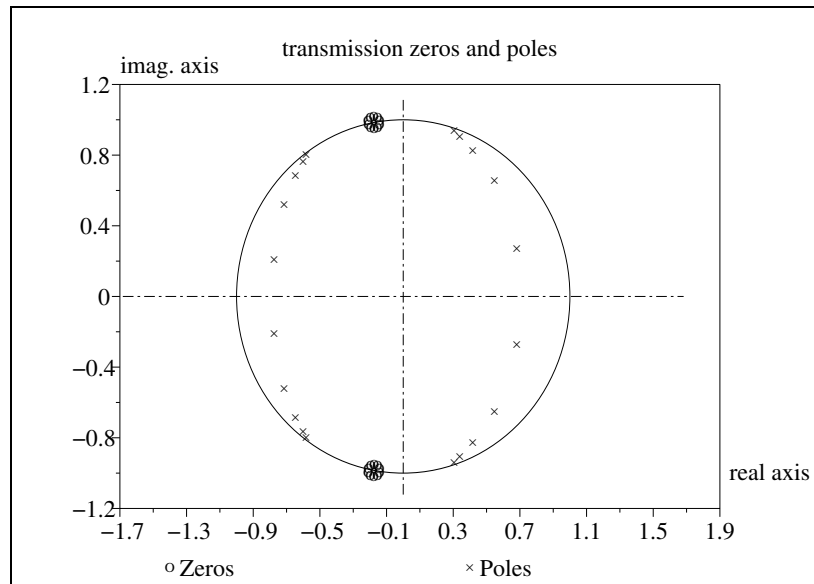


Figura 4.3: Grafico poli/zeri del filtro di Chebyshev tipo I

#### 4.2.2 Filtro di Chebyshev - tipo II

Creiamo un filtro di Chebyshev passa-basso di ordine 8 con frequenza di taglio 0,15 e frequenza di stop 0,25.

Il diagramma dei poli/zeri e della risposta in frequenza può essere ottenuto in Scilab con i seguenti comandi

```
// filtro CHSBYSCEV tipo II
[hz]=iir(8,'lp','cheb2',[.15 .25],[0.1 0.1]);
[hzm,fr]=frmag(hz,1000);
xbasc();xset('window',0);
// grafico risposta in frequenza
```

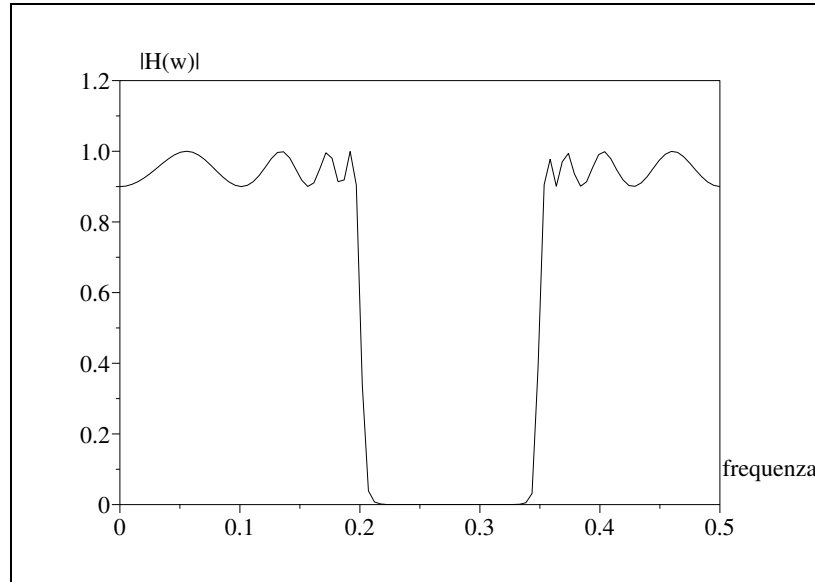


Figura 4.4: Risposta in frequenza del filtro di Chebyshev tipo I

```
plot2d(fr,hzm,rect=[0,0,.5,1.05]);
xtitle(tit="",xax="frequenza",yax="|H(w)|");
// grafico poli/zeri
xset('window', 1);
plzr(hz);
```

Otteniamo così il risultato di figura 4.6 e figura 4.5.

### 4.3 Filtri ellittici

I filtri ellittici sono caratterizzati da un ottimo guadagno sia in banda passante che in banda di transizione ma da una fase non lineare di pessima qualità.

La forma generale del modulo del filtro ellittico è del tipo

$$\frac{1}{\sqrt{1 + \epsilon^2 R_n^2(\omega)}}$$

Creiamo un filtro ellittico passa-basso di ordine 4 con frequenza di taglio 0,15 e frequenza di stop 0,25.

Il diagramma dei poli/zeri e della risposta in frequenza può essere ottenuto in Scilab con i seguenti comandi:

```
// filtro ELLITTICO
```



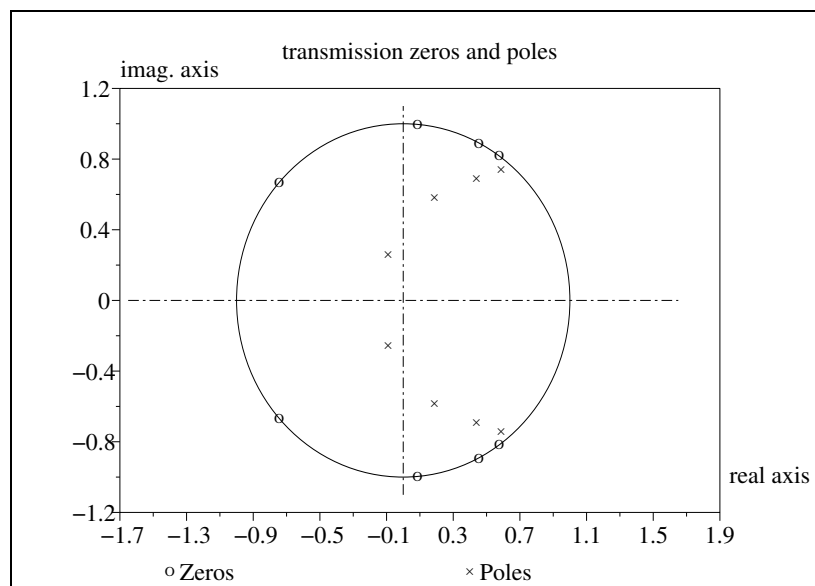


Figura 4.5: Grafico poli/zeri del filtro Chebyshev tipo II

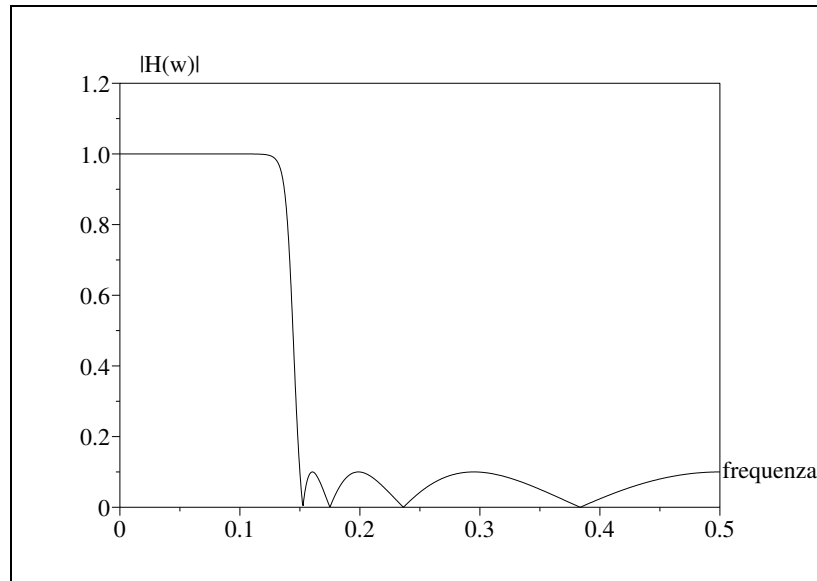


Figura 4.6: Risposta in frequenza del filtro Chebyshev tipo II

```

[hz]=iir(4,'lp','ellip',[.15 .25],[0.1 0.1]);
[hzm,fr]=frmag(hz,1000);
xbase();xset('window',0);
// grafico risposta in frequenza
plot2d(fr,hzm,rect=[0,0,.5,1.1]);
xtitle(tit="",xax="frequenza",yax="|H(w)|");
// grafico poli/zeri
xset('window', 1);
plzr(hz);

```

Otteniamo così il risultato di figura 4.8 e figura 4.7.

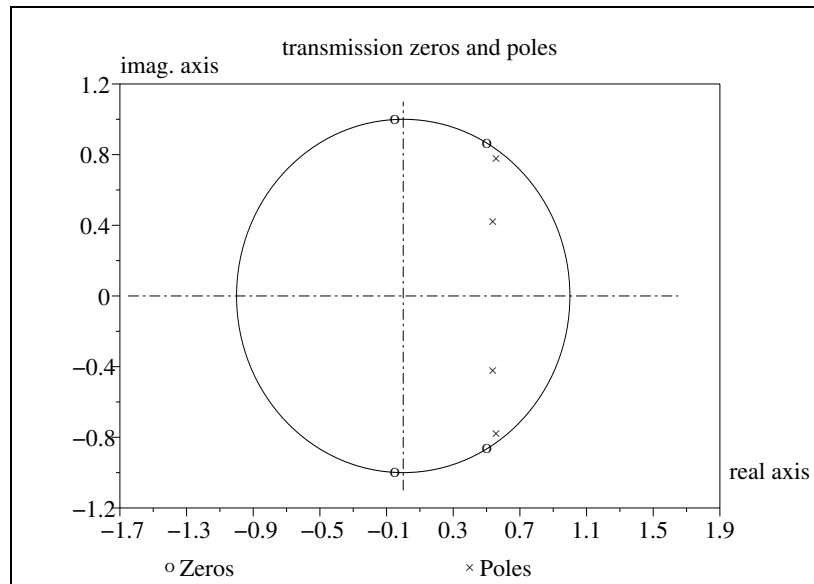


Figura 4.7: Grafico poli/zeri del filtro Ellittico

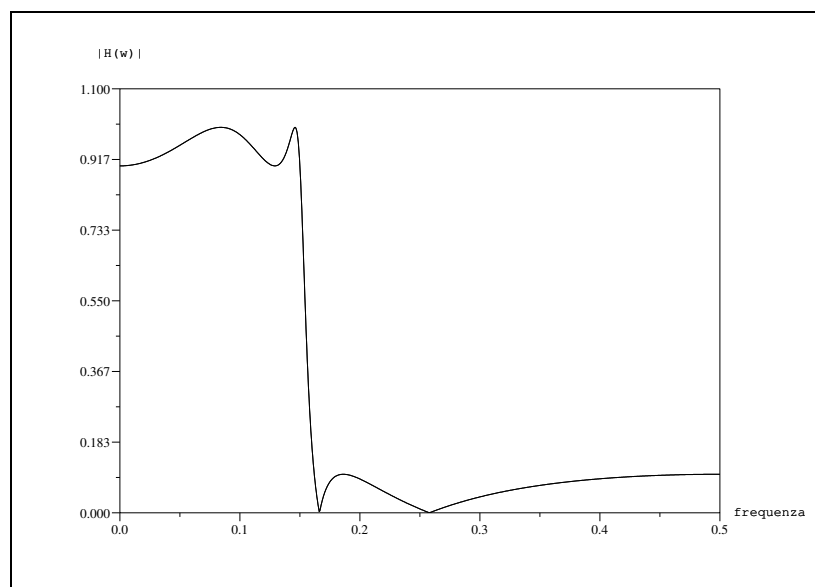


Figura 4.8: Risposta in frequenza del filtro Ellittico

# Appendice A

## Introduzione a Scilab

### A.1 I fondamenti

Scilab è un applicativo scientifico Open Source per il calcolo numerico.

I programmi vengono scritti ed interpretati riga dopo riga, non è quindi necessario definire a priori il tipo delle variabili utilizzate. Scilab è *case sensitive*, bisogna quindi fare attenzione alla distinzione tra lettere maiuscole e minuscole.

Benchè il programma supporti i cicli (for, while, ...), è ottimizzato per i calcoli matriciali ed è quindi conveniente tenerne in considerazione in fase di impostazione del calcolo.

### A.2 La riga di comando

Dalla riga di comando è possibile ottenere aiuto digitando **help** e il nome della funzione desiderata; se non ricordiamo il nome esatto, il comando **apropos** permette di ottenere un elenco di funzioni correlate.

Scilab permette di scorrere le precedenti istruzioni tramite l'uso delle frecce della tastiera.

I commenti nel codice sono preceduti da `//`.

In un numero decimale, il carattere di separazione è il punto.

Se non si desidera vedere i risultati di un inserimento, è possibile inserire un `;` alla fine dell'istruzione.

Le costanti predefinite (pi, i, e, ...) sono precedute da `%`.

### A.3 Vettori e matrici

Vettori e matrici possono essere definiti in vari modi:

- esplicitamente, racchiudendo gli elementi tra parentesi quadre e sepa-

randoli con il carattere ‘,’ per i vettori riga (può essere omesso) e ‘;’ per i vettori colonna;

- indicando l’elemento iniziale, l’elemento finale e il passo, con il seguente costrutto: (`valore_iniziale,passo,valore_finale`), il passo unitario e le parentesi possono essere omesse;
- per creare vettori e le matrici costituite esclusivamente da simboli ‘1’ o ‘0’, possono essere utilizzati i comandi `ones(righe,colonne)` o `zeros(righe,colonne)`;
- le matrici con gli elementi unitari sulla diagonale possono essere create con il comando `eye(a,b)`

Il carattere ‘`’` è l’operatore di trasposizione. L’operazione di moltiplicazione (rispettivamente divisione) elemento per elemento è rappresentata da `.*` (`./`), mentre vengono utilizzati i comuni operatori per le altre operazioni (`+`, `-`, `*`, `/`), ed `e` per le potenze di 10.

Alcune comuni funzioni utili:

- `sqrt()` restituisce la radice quadrata
- `sin()`, `cos()`, `tan()` restituiscono il seno, coseno, tangente
- `asin()`, `acos()`, `atan()` restituiscono le funzioni inverse delle precedenti
- `exp()`, `log()` restituiscono l’esponenziale e il logaritmo
- `10\^()`, `log()` restituiscono le funzioni potenza e algoritmo di 10
- `real()`, `imag()` restituiscono le parti reali e immaginarie di un complesso
- `abs()`, `phasemag()` restituiscono il modulo e la fase di un complesso

Alcune funzioni utili per le matrici:

- la funzione `matrix(v,n,m)` restituisce una matrice (n,m) a partire dalla matrice v (qualunque dimensioni abbia)
- `size(A)` restituisce il numero di righe e colonne di A
- `length(A)` restituisce il numero di elementi di A
- `sum(A)` restituisce la somma di tutti gli elementi di A
- `sum(A, ‘r’)` restituisce un vettore riga i cui elementi sono le somme delle colonne corrispondenti di A

- `sum(A, 'c')` restituisce un vettore colonna i cui elementi sono le somme delle righe corrispondenti di A
- `triu(A)` restituisce una matrice i cui elementi sopra la diagonale (compresa) sono uguali a quelli della matrice A, nulli gli altri
- `tril(A)` restituisce una matrice i cui elementi sotto la diagonale (compresa) sono uguali a quelli della matrice A, nulli gli altri
- `conj(A)` restituisce la matrice complessa coniugata di A

Il valore di un vettore può essere ottenuto in vari modi:

- `s(n)` restituisce l'n-esimo elemento (l'indice parte da 1)
- `s(a:p:b)` restituisce gli elementi tra gli indici a e b con passo p
- `s(a,b)` restituisce l'elemento di posizione (a,b)
- `s(:, b)` restituisce tutta la colonna b

## A.4 Visualizzazione dei grafici

Le funzioni `xbasec([numero_finestra])` e `xset("parametro",valore)` inizializzano la finestra in cui verrà disegnato il grafico. Comuni parametri di `xset` sono `font size` e `window` (per attivare una finestra con il numero indicato), chiamando `xset()` senza parametri è possibile impostarli usando un'interfaccia grafica.

`plot2d(var_indipendente, var_dipendente, parametri)` disegna il grafico bidimensionale corrispondente. Non esplicitando la variabile indipendente, il grafico verrà disegnato utilizzando gli indici sull'asse delle ascisse. Alcuni parametri comunemente usati:

- `rett=[minY, minY, maxX, maxY]` permette di restringere la rappresentazione del grafico nel dominio e nel codominio indicato
- `style(x)` definisce lo stile del tratto (i valori positivi di x rappresentano i colori, i valori negativi definiscono il tratteggio, la punteggiatura, l'uso del carattere 'x' invece del tratto continuo)
- `logflag"xy"` definisce il tipo di scala sull'asse delle ascisse e delle ordinate, con possibilità di scelta tra N (normale) o L (logaritmica)
- `axesflags=[n]` definisce le caratteristiche degli assi

In alcune circostanze, si ottengono migliori risultati utilizzando `plot2d2` (con funzioni costanti a tratti) o `plot2d3` (in presenza di rette del tipo  $x = k$ , le altre funzioni di disegno 2D tracciano un triangolo, mentre questa traccia una riga netta). Altre funzioni comunemente utilizzate:

- `xtitle("titolo","asse_ascisse","asse_ordinate")` permette di inserire il titolo e le legende sugli assi del grafico per chiarirne il significato.
- `xstring(x_bg, y_bg, [ "line 1"; "line 2"; ....])` permette di aggiungere testo nella posizione indicata.
- `xarrows([xp1,xa1,xp2,xa2],[yp1,ya1,yd2,ya2])` permette di aggiungere frecce al grafico (indicated le coordinate di partenza e di arrivo).
- `xarc(x,y,w,h,a1,a2)` permette di disegnare un'ellisse contenuta nel rettangolo di vertice superiore sinistro ( $x,y$ ), larghezza ( $w$ ), altezza ( $h$ ) e nel settore definito dagli angoli  $a1$  e  $a2$  (espressi in 64-esimi di grado)
- `xgrid(n)` permette di aggiungere una griglia, e il parametro  $n$  ne definisce lo stile.
- `xsetech([x0,y0,largh,alt])` imposta la visualizzazione del grafico nel rettangolo con vertice sinistro inferiore di coordinate ( $x0,y0$ ), larghezza  $largh$  e altezza  $alt$  (i parametri  $x0$ ,  $y0$ ,  $largh$  e  $alt$  sono espressi con valori tra 0 e 1).

## A.5 Salvataggio e ripristino dell'ambiente di lavoro

Scilab mette a disposizione le funzioni `save` e `load` per salvare e ripristinare successivamente le variabili create in un file binario.

Se invece il codice viene memorizzato in un file esterno, è possibile eseguirlo con la funzione `exec`.

## A.6 Programmazione

Scilab mette a disposizione i costrutti tipici di altri linguaggi di programmazione, ad esempio i cicli `for`, `while`, `if...then...else`, `select...box`. Normalmente è computazionalmente più conveniente sostituire il ciclo con una matrice algebrica.

Scilab permette infine di definire nuove funzioni secondo la sintassi: `funzione[y1, y2 ... yn] = nome_della_funzione(x1, x2 ... xm)`

## A.7 Polinomi e funzioni di trasferimento

Un polinomio viene descritto in due possibili modi:

- `poly([radici], 'p')` se sono note le radici
- `poly([coeff], 'p', 'c')` se sono noti i coefficienti (che devono essere elencati a partire da quello di grado 0)

La funzione `roots(p)` consente di ottenere le radici del polinomio  $p$ , mentre `horner(P,x)` permette di valutare il polinomio quando la variabile in cui è definito assume valore  $x$ .

## A.8 Funzioni per l'analisi numerica dei segnali

Ecco infine una rassegna delle più comuni funzioni utilizzate nell'analisi numerica dei segnali. Per la sintassi precisa fare riferimento al comando `help <nome_comando>`.

- `syslin` permette di definire un sistema lineare dati due polinomi (spesso il numeratore ed il denominatore della funzione di trasferimento) o la funzione di trasferimento stessa. Il sistema lineare può così essere manipolato come una comune matrice (concatenazione, estrazione, trasposta, moltiplicazione, ...)
- `plzr` visualizza un grafico poli/zeri della funzione di trasferimento passata in ingresso come un sistema lineare
- `roots` restituisce le radici del polinomio in ingresso
- `trzeros` restituisce gli zeri del sistema lineare in ingresso
- `freq(num,den,f)`, data la funzione di trasferimento del filtro (espressa come numeratore e denominatore) e il segnale in ingresso, restituisce la risposta in frequenza come matrice reale o complessa
- `frmag(num[,den],num_punti)` restituisce la risposta in frequenza del filtro in una matrice di due colonne: la colonna dei punti nel dominio frequenziale su cui è valutata la magnitudine della risposta in frequenza, e la colonna dei valori corrispondenti
- `bode` restituisce il grafico di Bode (magnitudine e fase) della risposta in frequenza del sistema lineare in ingresso
- `gainplot` agisce come `bode`, ma restituisce solo la magnitudine
- `freq(P,x)` valuta il polinomio  $P$  quando la variabile ha valore  $x$  ( $x$  può essere un numero reale o un polinomio)
- `wfir(tipo_fil,ordine,freq_taglio,tipo_fin,parametri_fin)` restituisce i coefficienti del filtro FIR a fase lineare realizzato con il metodo delle finestre



- `eqfir(num_punti_uscita,bande,magnitudine_bande,peso_bande)`  
restituisce i coefficienti del filtro FIR multibanda a fase lineare realizzato con il metodo di approssimazione minimax
- `iir(n,tipo,design,freq_taglio,errori)` restituisce i coefficienti del filtro IIR realizzato secondo le specifiche fornite

## Appendice B

# Introduzione a Scicos

SCICOS (SCIlab Connected Object Simulator) è una libreria di Scilab che permette di simulare sistemi dinamici, sia continui che discreti e avvalendosi di un editor grafico permette di costruire modelli tramite l'interconnessione di blocchi.

I blocchi rappresentano funzioni base o funzioni definibili dall'utente e agiscono come un sottosistema che viene attivato da un input e fornisce come output un segnale dipendente dall'ingresso e dal suo stato.

In Scicos ad ogni segnale è associato un insieme di indici di tempo, detti istanti di attivazione, in cui evolve il segnale (per istanti non di attivazione il segnale rimane costante).

I segnali vengono generati da blocchi pilotati da segnali di attivazione che portano il blocco ad adeguare la sua uscita in funzione dell'ingresso e del suo stato interno.

### B.1 L'editor di Scicos

Per avviare l'editor grafico attivando la finestra principale di Scicos è necessario digitare `scicos()` dall'editor di Scilab; come si vedrà l'interfaccia grafica è intuitiva e permette di creare facilmente i diagrammi dei sistemi. Per costruire un modello in Scicos si eseguiranno tipicamente le seguenti operazioni:

- aprire il menu **edit** e cliccare sulla voce **palettes**: verranno mostrate diverse categorie di blocchi che rappresentano le componenti tipiche dei sistemi
- copiare i blocchi nella finestra di Scicos, cliccando sul componente desiderato e posizionandolo sulla finestra principale con un altro click
- connettere le porte di input e output dei blocchi, cliccando semplicemente sulle porte che si desiderano collegare. Nel caso risultasse

necessario biforcare l'uscita di un blocco per connetterla a più componenti si deve creare un nuovo link, scegliendone l'origine sul link da biforcare

- inserire nei diagrammi almeno un blocco per poter visualizzare o salvare gli esiti della simulazione, ad esempio `Mscope` o `scrivi su file`

## B.2 Blocchi base

I blocchi base disponibili in Scicos sono di tre tipi:

- **Regular Basic Block**  
possono rappresentare variabili continue  $x$  o discrete  $z$ ;
- **Zero Crossing Basic Block**  
generano output solo se almeno un input regolare cambia segno. L'evento generato può dipendere dalla combinazione degli eventi che hanno cambiato segno;
- **Synchro Basic Block**  
blocchi di sincronizzazione; possono avere più input o output di attivazione, ma hanno un solo input regolare; appena ricevono in input un segnale di attivazione generano un segnale di attivazione in output scegliendo in base all'input regolare.

Ogni blocco può avere due tipi di porte sia di input sia di output: porte per input regolari, per input di attivazione, per output regolari e per output di attivazione.

## B.3 Impostazione dei parametri delle componenti

I parametri dei blocchi possono essere modificati così da simulare fedelmente il sistema da realizzare. Per modificarli, è sufficiente cliccare due volte sul componente.

## B.4 Simulazione

Una volta completato lo schema del sistema e dopo aver impostato i parametri di ogni blocco, selezionando l'opzione `run` dal menù `simulate` il programma comincia la simulazione in una nuova finestra. La simulazione può essere interrotta premendo il pulsante `stop` nella finestra principale.

## B.5 Altre funzionalità

- salvare e caricare programmi Scicos
- modificare l'aspetto dei blocchi
- esportare e stampare i diagrammi e le simulazioni di Scicos
- interagire con codice C e Fortran
- interagire con file audio
- visualizzare l'help riguardante i blocchi (selezionare **help** e poi cliccare sul componente desiderato)
- modularizzare sistemi complessi in sub-sistemi, così da semplificare lo schema generale e favorire la riusabilità delle componenti

## B.6 Esempio: simulazione di un filtro passa-basso

Al fine di dimostrare un possibile impiego di Scicos, realizziamo una simulazione di un segnale composto da due sinusoidi filtrato attraverso un filtro passa-basso:

1. inseriamo i due generatori di sinusoidi e li colleghiamo ad un sommatore, ottenendo così il segnale di input
2. definiamo la funzione di trasferimento del filtro come rapporto tra due polinomi in  $z$
3. aggiungiamo un clock per attivare il sistema ad intervalli stabiliti
4. poniamo in ingresso ad un oggetto **MScope** il segnale in ingresso prima e dopo il passaggio attraverso il filtro

Avviando la simulazione del sistema, si ottiene l'uscita in figura B.2.

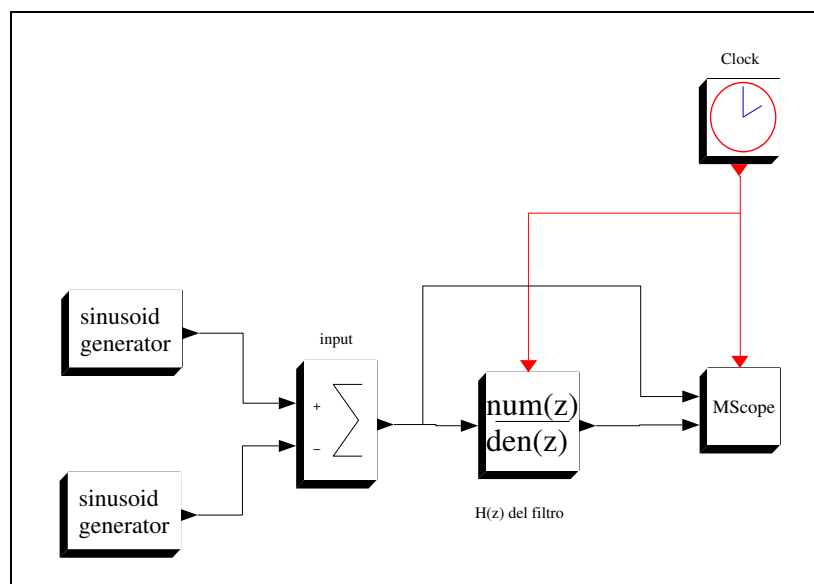


Figura B.1: Esempio di schema Scicos

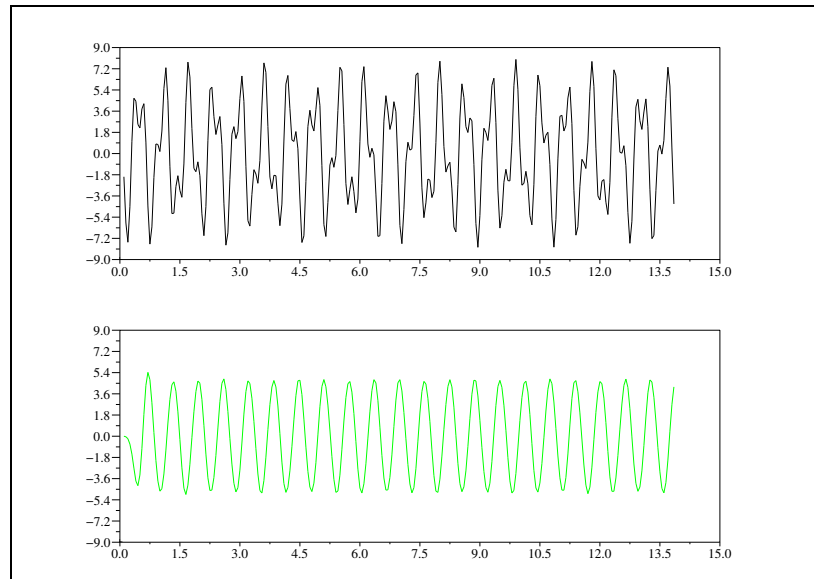


Figura B.2: Segnale in input e segnale filtrato

# Bibliografia

- [1] A. Bertoni, G. Grossi - *Dispensa del corso di elaborazione numerica dei segnali* - Università degli Studi di Milano, Dipartimento di Informatica - A.A. 2001-2002
- [2] R. Kuc - *Introduction to Digital Signal Processing* - McGraw-Hill - 1988
- [3] F. Rocca - *Elaborazione numerica dei segnali* - Edizioni CUSL Milano - 1999
- [4] Online courses at Rice University (Houston, US): [http://cnx.rice.edu/content/browse\\_courses](http://cnx.rice.edu/content/browse_courses)
- [5] Online courses at Bores Signal Processing (UK): [http://www.bores.com/index\\_online.htm](http://www.bores.com/index_online.htm)
- [6] Online material at Strathclyde University (Glasgow, Scotland): <http://www.spd.eee.strath.ac.uk/interact/>
- [7] J. O. Palacios - *An introduction to the treatment of neurophysiological signals using Scilab*: <http://www.neurotraces.com/scilab/scilab2/index.html> - 2001
- [8] D. Rabasté - *Traitement numérique du signal avec Scilab*: [http://artemis.univ-mrs.fr/iufin-genelec-forum/SCILAB/introduction\\_pages\\_scilab.htm](http://artemis.univ-mrs.fr/iufin-genelec-forum/SCILAB/introduction_pages_scilab.htm) - Institut Universitaire de Formation des Maîtres de l'Académie d'Aix Marseille - Marseiglia (FR)
- [9] S. W. Smith - *The Scientist and Engineer's Guide to Digital Signal Processing*: <http://www.dspguide.com> - 2002
- [10] Scilab documentation: <http://scilabsoft.inria.fr/doc.html>

Questo documento è stato scritto in L<sup>A</sup>T<sub>E</sub>X e revisionato in data 29 giugno 2004.