

MATLAB 5.0

Il programma MATLAB, distribuito dalla MATH WORKS Inc., è stato ideato dai ricercatori dell'università di New Mexico e di Standford, e si è imposto a livello mondiale come strumento fondamentale per la simulazione e l'analisi dei sistemi lineari e non lineari, continui e discreti utilizzati sia in ambito universitario e didattico che industriale.

Il programma dispone di diversi **tool**, tra cui:

- **MATH**: costituito da numerose funzioni di calcolo sia di algebra che di analisi matematica e di una potente gestione delle matrici e della grafica 2D e 3D.
- **CONTROL SYSTEM TOOLBOX**: Insieme di funzioni e comandi specializzati per l'analisi dei sistemi di controllo a catena aperta e chiusa tempo continuo e tempo discreto.
- **SIMULINK**: integrato in Matlab opera in modalità grafica e dispone di una vasta gamma di blocchi funzionali che, opportunamente connessi, consentono la simulazione e la rappresentazione grafica nel dominio del tempo e in quello armonico dei sistemi di controllo.

Dopo aver lanciato il programma appare, nella versione Education, il prompt **EDU>>**.

Si digita l'operazione che si intende eseguire e si preme il tasto INVIO. Questo modo di operare è detto *immediato* e non consente la memorizzazione della sequenza di operazioni digitate. Si possono solo ripetere i comandi attivando i tasti freccia.

Se si desidera memorizzare il lavoro, per operare delle modifiche e/o correzioni, è necessario scrivere le operazioni in un file di testo utilizzando l'editor interno a Matlab o un qualsiasi editor di testi come il Blocco Note di Windows. Il file deve essere salvato con l'estensione **.m** L'editor di Matlab si richiama selezionando New M-file dal menu File. Per scegliere un altro editor selezionare Preferences dal menu File. Per eseguire un M-file è necessario richiamarlo con il comando Run Sript del menu File.

Di seguito si descrivono i comandi e le funzioni fondamentali di Matlab. Per un'analisi più approfondita si rimanda alla guida in linea.

INSERIMENTO INTERVALLI

$X = x_{min}$: passo: x_{max} Esempio $x = 0:0.1:10$ oppure $x = [0:0.1:10]$;

Il ; dopo il comando non visualizza i dati

`inspace(Vin, Vfin, NoPunti)` Esempio `inspace(-pi, pi, 4)` se NoPunti è omissso vale 100

`logspace (Vin, Vfin, NoPunti)` Esempio `logspace (1, 2, 100)` definisce 100 punti nell'intervallo 10 – 100. Per default NoPunti = 50.

Formato dei dati

Il programma opera internamente sui dati in formato *doppia precisione*. Il formato di uscita, prevede, normalmente, solo 5 cifre significative. Per modificare il formato di uscita si deve agire sul *menu file / preferences*.

eps fornisce la precisione della macchina

FUNZIONI ELEMENTARI

| | |
|--------------------|--|
| abs | Valore assoluto |
| acos, acosh | Arcocoseno e arcocoseno iperbolico |
| acot, acoth | Arcocotangente e arcocotangente iperbolica |
| acsc, acsch | Arcocosecante e arcocosecante iperbolica |
| angle | Fase angolo |
| asec, asech | Arcosecante e arcosecante iperbolica |
| asin, asinh | Seno e seno iperbolico inverso |
| atan, atanh | Arcotangente e arcotangente iperbolica |
| atan2 | Arcotangente nei 4 quadranti |
| conj | Complesso coniugato |
| cos, cosh | Coseno e coseno iperbolico |
| cot, coth | Cotangente e cotangente iperbolica |
| csc, csch | Cosecante e cosecante iperbolica |
| exp | Esponenziale |
| gcd | M.C.D. |
| imag | Parte immaginaria di un numero complesso |
| lcm | m.c.m. |
| log | Logaritmo naturale |
| log2 | Logaritmo in base 2 |
| log10 | Logaritmo in base 10 |
| mod | Modulo |
| pow2 | Esponenziale in base 2 |
| rat | Espansione razionale |
| rats | Approssimazione razionale |
| real | Parte reale di un numero complesso |
| rem | Resto di una divisione |
| round | Round to nearest integer |
| sec, sech | Secante e secante iperbolica |
| sign | Segno |
| sin, sinh | Seno e seno iperbolico |
| sqrt | Radice quadrata |
| tan, tanh | Tangente e tangente iperbolica |

Esempio

`rats(9.22)` fornisce 461/50
`rats/(1 -1/2 +1/3 -1/4 + 1/5)` fornisce 57/60 (60 è il m.c.m.)

operatore `rat`

`[n,d] = rat(x, tol)` la tolleranza per default è 10^{-6}

Esempio

`rat(sqrt(2))` fornisce $1 + 1/(2 + 1/(2 + 1/(2 + \dots$

Numeri complessi

I numeri complessi usano indifferentemente l'operatore *i* oppure *j*

Esempi: $z = 2 + j3$ oppure $z = 2 + j*3$ oppure $z = 2 + i*3$ oppure $z = 2 + i3$

Attenzione a non usare *i* o *j* in altre operazioni come i *cicli for*

Per ripristinare la variabile al valore di default usare il comando **clear** ad esempio: `clear i`

Cambio coordinate

`cart2pol` cartesiane -----> polari
`pol2cart` polari-----> cartesiane
`cart2sph` cartesiane-----> sferiche

Esempi `[teta, ro] = cart2pol (x, y)`
 `[teta, ro, z] = cart2pol (x, y, z)`
 `[fi, teta, ro] = cart2sph (x, y, z)`

OPERATORI

`==` uguale

`-` diverso

`&` AND

`|` OR

`xor` operatore XOR

`~` NOT Esempio $2 + 2 \sim = 0$ (FALSO)

Operazioni logiche

`any (x)` con x vettore binario. Fornisce l'OR tra gli elementi

`all (x)` con x vettore binario. Fornisce l'AND tra gli elementi

Esempio: `all(A < 4)` fornisce 1 solo se tutti gli elementi sono < 4

FUNZIONI LOGICHE

C = bitand(13,27)
Fornisce: C = 9

%C= 13 AND 27 ovvero:13 vale 01101; 27 vale11011
% 9 corrisponde a 01001

Altre funzioni:

C = bitcmp(A,n)
C = bitor(A,B)
C = bitxor(A,B)
C = bitget(A,bit)
C = bitshift(A,n)

Complemento a 1 della parola A su n bit
Funzione OR
Funzione XOR
Fornisce il valore del bit della parola A nella posizione bit
Produce uno scorrimento a destra (n>0) o a sinistra (n<0) dei bit di A di ln posizioni

C = strcmp(a,b)

Restituisce 1 se le stringhe **a** e **b** sono uguali, altrimenti C = 0.
Dove a = 'stringa1' e b = 'stringa2'

MATRICI

A = [1, 2, 3; -3, 6, 0]
a = A(i, j)
B = A'
B = inv (A) oppure = A^(-1)
max (X)
min (X)
[Y, I] = max (A)
[Y, I] = sort(A)
sum (X)
mean (X)
rank (A)
det (A)
poly (A)
trace (A)
i = find(A)
[i,j] = find(A)
eye(m,n)
zeros(m,n)
ones(m,n)
diag(A)
rot90(A)
[m,n] = size(A)

Matrice con 2 righe e 3 colonne
Fornisce l'elemento a_{ij}
Matrice trasposta
Matrice inversa
Fornisce il max elemento del vettore
Fornisce il min elemento del vettore
Y vettore dei massimi e I l'indice dell'elemento
Y vettore ordinato e I indice degli elementi
Somma gli elementi del vettore
Valore medio
Massimo numero di righe o colonne indipendenti
Determinante
Fornisce i coefficienti del polinomio caratteristico.
Somma degli elementi della diagonale principale
Fornisce gli indici degli elementi non nulli del vettore
Fornisce gli indici non nulli riga-colonna della matrice
Matrice identità
Matrice contenente tutti 0
Matrice contenente tutti 1
Vettore della diagonale principale
Ruota la matrice di 90° in senso antiorario
Fornisce le dimensioni della matrice

Risoluzione di un sistema in forma matriciale: [A]*[X] = [B]

$X = A \setminus B$

Esegue l'operazione $X = A^{-1} * B$

Esempio

A = [1,2,3;-3,4,5;1,0,2]

A =

| | | |
|----|---|---|
| 1 | 2 | 3 |
| -3 | 4 | 5 |
| 1 | 0 | 2 |

B = [1;-5;6]

B =

| |
|----|
| 1 |
| -5 |
| 6 |

X = A \ B

X =

| |
|---------|
| 0.8889 |
| -3.7778 |
| 2.5556 |

Esempio

```
A = [1,3,0; 4,5,1]
[i,j] = find(A>2)
```

Genera $i = (2 \ 1 \ 2)$ e $j = (1 \ 2 \ 2)$ che indica che gli elementi a_{21} , a_{12} , a_{22} sono >2

Esempio

Determinare le radici dell'equazione $|\sin t - 0.5| = 0$ con una approssimazione del 2%

```
t = [0:0.005:20];
y = sint;
i = find(abs(y - 0.5) < 0.02)
```

Operazioni tra vettori

Sia $x = (x_1, x_2, x_3)$ e $y = (y_1, y_2, y_3)$

Prodotto scalare (è un numero) $x \cdot y = x_1 y_1 + x_2 y_2 + x_3 y_3$
Per ottenere il vettore prodotto degli elementi si deve scrivere:

$x \cdot y$ fornisce come risposta il vettore $(x_1 y_1, x_2 y_2, x_3 y_3)$
Analogamente per gli operatori \cdot / \cdot^{\wedge}

STRINGHE

```
num2str(N)
str2num(S)
```

Trasforma il numero N in stringa di caratteri
Trasforma la stringa S = 'stringa' nel numero corrispondente

Tabellare i valori di una funzione

Esempio Calcolo di 50 valori di $\log(x)$ nell'intervallo 1-5:

```
x=[1:0.1: 5];
y=log(x) ;
[x,y]
```

Esempio Calcolo di 45 valori di $y = e^{3t} \cdot \sin(5\pi t)$ nell'intervallo $[-2, +2]$

```
t=linspace[-2,2,45];
y=exp(3*t).*sin(5*pi*t); %Si noti il formalismo .* per il prodotto di funzioni
[t y]
```

Esempio

Rappresentare $F(s) = (3s^2 + 5s + 7) / (s^3 + 5s^2 + 7s + 12)$ per $s = j\omega$ e $\omega = [10^{-2} \div 10^2]$

```
Omega = logspace(-2,2);
```

```

s=j*omega
x1=3*s.^2+5*s+7;
x2=s.^3+s.^2+7*s+12      %Si noti il formalismo .* per il prodotto di funzioni
x=x1./x2;
x=abs(x);
[s x]

```

Polinomi

Sono vettori riga contenenti i coefficienti delle potenze decrescenti.

Assegnati i vettori $a = [1 \ 2 \ 3]$; $b = [4 \ 5 \ 6]$; $p = [1, -2, 5]$;

Operazioni fondamentali

Prodotto $c = \text{conv}(a,b)$

Divisione $[q,r] = \text{deconv}(a,b)$ $q = \text{quoziente}; r = \text{resto}$

Radici di un polinomio p $\text{roots}(p)$

Valore in un punto $\text{polival}(p,x)$ $\text{calcola } p(x)$

Derivate di un polinomio
 $q = \text{polyder}(p)$ $\text{derivata di un polinomio}$
 $q = \text{polyder}(a,b)$ $\text{derivata del prodotto } a*b$
 $[q,d] = \text{polyder}(a,b)$ $\text{derivata del rapporto } a/b$

Note le radici di un polinomio per ottenere il polinomio generatore: $p = \text{poly}(r)$

Esempio

```

r = [-3-2*j , -3+2+j, -5]
p=poly(r)

```

Esempio

```

disp('Inserimento di un polinomio. Esempio: x^2 + 2*x + 3')
polinomio='x^2 + 2*x + 3'
pause
disp('Radici di un polinomio noti i coefficienti' p=3x^2+4x-2)
p=[3 4 -2]
radici=roots(p)
pause
disp('Risoluzione di una equazione scritta in forma naturale')
f='2*x^2 + 4*x -1'
radici=solve('2*x^2 + 4*x -1','x')
disp('Restituisce il polinomio note le radici')
polin=poly(radici)
pause
disp('Prodotto tra due polinomi')
p1=[1 2 3]; p2=[4 5 6];
prodotto=conv(p1, p2)

```

Zeri di una funzione

Oltre alla funzione **roots** gli zeri di una funzione si possono calcolare con l'istruzione

fzero (f, x)

Esempio

fzero('sin',3) fornisce 3.1416

Funzione expand

Sviluppa le operazioni algebriche

Esempi

| | |
|------------------------------|--|
| expand((x-2)*(x-4)) | fornisce: $x^2-6*x+8$ |
| expand(cos(x+y)) | fornisce: $\cos(x)*\cos(y)-\sin(x)*\sin(y)$ |
| expand(exp((a+b)^2)) | fornisce: $\exp(a^2)*\exp(a*b)^2*\exp(b^2)$ |
| expand(log(a*b/sqrt(c))) | fornisce: $\log(a)+\log(b)-1/2*\log(c)$ |
| expand([sin(2*t), cos(2*t)]) | fornisce: $[2*\sin(t)*\cos(t), 2*\cos(t)^2-1]$ |

Interpolazione polinomiale

Dati 2 vettori x e y si ha:

$p = \text{polyfit}(x, y, n)$

fornisce un polinomio $p(x)$ di grado n che interpola le coppie (x_i, y_i)

$y_i = \text{interp1}(x, y, x_i, \text{'metodo'})$

dati due vettori x e y e un vettore x_i di ascisse la funzione fornisce una matrice y_i le cui colonne interpolano le rispettive colonne di y sulle stesse x_i

Il parametro metodo può essere: **linear** (per default), **spline**, **cubic**

Il vettore x deve essere monotono crescente; per cubic devono essere anche equidistanti

Risoluzione di equazioni e sistemi lineari

Esempio

Equazioni

solve('a*x^2 + b*x + c') Produce:

$$\left[\frac{1}{2/a}*(-b+(b^2-4*a*c)^{1/2}), \right. \\ \left. \frac{1}{2/a}*(-b-(b^2-4*a*c)^{1/2}) \right]$$

solve('a*x^2 + b*x + c','b') Produce:

$$-(a*x^2+c)/x$$

Sistema di 2 equazioni in due incognite
 solve('x + y = 1','x - 11*y = 5') Produce :

$$y = -1/3, x = 4/3$$

Limiti

Esempi

syms x h; => definisce le variabili in uso
 limit(sin(x)/x) => 1
 limit(1/x,x,0,'right') => inf
 limit(1/x,x,0,'left') => -inf
 limit((sin(x+h)-sin(x))/h,h,0) => cos(x)

Derivata e Integrale

Esempio . Calcolare la derivata e l'integrale della funzione sin(x):

f= sin(x);
 y = diff(f) fornisce la derivata della funzione f
 y = int(f) fornisce l'integrale della funzione f
 y = int(f,a,b) fornisce l'integrale definito

Esempio

```
syms x
f = 4*x^5 + sin(x) %funzione da derivare
disp('Calcolo della derivata della funzione f') %Commento
y=diff(f) %derivata
pretty(y) %"aggiusta" il risultato
disp('Calcolo integrale della funzione f') %Commento
z = int(f) %integrale
pretty(z) %"aggiusta" il risultato
S = int(f,1,2) % integrale definito tra 1 e 2
```

Il software produce la seguente risposta

f =
 4*x^5+sin(x)
 Calcolo della derivata della funzione f
 y =
 20*x^4+cos(x)

$20 x^4 + \cos(x)$

Calcolo integrale della funzione f
 z =
 2/3*x^6-cos(x) $2/3 x^6 - \cos(x)$

S =
 42-cos(2)+cos(1)

Equazioni differenziali

Esempio

Assegnata l'equazione differenziale $y'' + 4y - 6 = 0$ si risolve:

```
dsolve(' D2y + 4*y - 6 = 0', 'x') %se si omette 'x' la risoluzione è nella variabile t
pretty(ans) %visualizza in formato tipicamente matematico
```

si ottiene

$$\frac{3}{2} + C1 \cos(2x) + C2 \sin(2x)$$

$$3/2 + C1 \cos(2 x) + C2 \sin(2 x)$$

Se sono assegnate delle condizioni iniziali, ad esempio: $y(1) = 0$ e $y(2) = -1$ si deve:

```
dsolve(' D2y + 4*y - 6 = 0', 'y(1) = 0', 'y(2) = -1', 'x')
```

si ottiene

$$\frac{3}{2} + \frac{5}{2} - 3 \cos(2) \cos(2x) - \frac{1}{2} \frac{(-3 + 5 \cos(2) + 6 \sin(2)^2)}{\sin(2)} \sin(2x)$$

Serie di Taylor

```
syms x
f = taylor(log(1+x));
ccode(f)
```

si ottiene

$$t0 = x - x^2/2 + x^3/3 - \text{pow}(x,4,0)/4 + \text{pow}(x,5,0)/5;$$

Time e dates

date fornisce la data corrente come stringa
 now fornisce la data corrente come numero
 calendar visualizza il calendario del mese corrente
 tic parte il clock interno
 toc fornisce il numero di secondi tra tic e toc

Operazioni su vettori

Assegnato un vettore $p = [2,3,5,-2]$

max(p) fornisce 5
 min(p) fornisce -2
 mean(p) fornisce il valore medio 1.8
 std(p) la standard deviation 2.5884
 sort(p) ordina fornisce : -2 1 2 3 5
 gradient(p,2) fornisce 0.5000 0.7500 -1.2500 -3.5000

Grafica

1) Dato un vettore $y = [0, 48, 2, \dots]$;

plot(y)

traccia il grafico della funzione con ordinate date dai valori del vettore y

plot ammette un parametro \rightarrow plot (y,'parametro'). I parametri sono

| <i>Tipo linea</i> | <i>Tipo punto</i> | <i>Colore</i> |
|-------------------|-------------------|------------------|
| Continua - | Punto . | Giallo y |
| Tratteggiata -- | Più + | Magenta m |
| Punteggiato ; | asterisco * | Ciano c |
| Tratto punto -. | Cerchio o | Rosso r |
| | Croce x | Bianco w |
| | | Nero k |
| | | Verde g |

Esempio plot[y, 'og'] \rightarrow cerchi verdi

Commenti Per inserire dei commenti sul grafico:

title ('titolo')
 xlabel ('ascisse')
 ylabel ('ordinate')
 grid; griglia. Comando di tipo ON/OFF. Anche xgrid e ygrid

Per inserire un testo in un punto generico di coordinate x,y
 text (x,y,' testo')

Inserimento del testo con il mouse gettext('testo')

Esempio

```
t = 0:0.05:4*pi;
y = sin(t);
plot(t,y,'r-') linea tratto/punto rosso
```

Esempio Funzioni con scale diverse su due distinte figure

```
t1 = 0:0.1:1; %1ª scala
y1 = exp(-10*t1) + sin(0.05*t1); %funzione f(t)=e-10t+sin(0.05t)

t2 = 0:200; %2ª scala 0÷200 con passo 1
y2 = exp(-10*t2) + sin(0.05*t2);
figure(1) plot (t1,y1)
figure(2) plot (t2,y2)
```

Grafici sovrapposti Funzioni con la stessa scala.

Esempio Grafico delle funzioni $y=\sin(t)$ e $y=\cos(t)$ per t compreso in $[0,2\pi]$

```
t=(0: 0.1: 2*pi)';
y=[sin(t), cos(t)];
plot(t,y)
```

Esempio Grafico di: $\sin(t)$ con t compreso in $[0,3]$ e di $\cos(t)$ con t compreso in $[1,4]$

```
t1=(0: 0.1: 3) '    y1=sin(t1)
t2=(1: 0.1: 4) '    y2=cos(t2)
plot (t1, y1, t2, y2)
% per 'congelare' il grafico usare:
axis(axis)
hold
```

Esempio Confrontare il grafico della funzione $\sin(t)$ con le funzioni $y=t$ e $Y=e^t$

```
t = 0: 0.1: 5
plot(t, sin(t))
axis (axis)
hold
plot (t,[t; exp(t)] )
```

Scalatura assi

| | |
|---|----------------------------------|
| axis([x_{min} y_{max} y_{min} y_{max}]) | |
| axis (axis) | “congela gli assi” |
| axis (auto) | “ripristina la scalatura “ |
| axis ('off') | “disattiva il tracciamento assi” |
| axis (on) | |
| axis ('square') | |
| axis ('equal') | |
| axis ('x y') | |
| semilog x | asse x in scala log |
| semilog y | asse y in scala log |
| loglog | sia x che y in scala log |
| polar(σ, ξ) | |
| bar | diagramma a barre |
| stairs | diagramma a gradini |
| hist | istogrammi |

Esempio

Istogramma con 20 barre con numeri casuali

```
n=100;
y = rand(n, 20);
hist (y, x)
```

Esempio

```
fplot ('funzione', [xmin xmax])
oppure
```

`[x y]=fplot('funzione', [x_min x_max])` restituisce i valori
`plot(x; y)` grafico funzione

Grafico di funzioni

Esempio

```
x=-10:0.01:10;
```

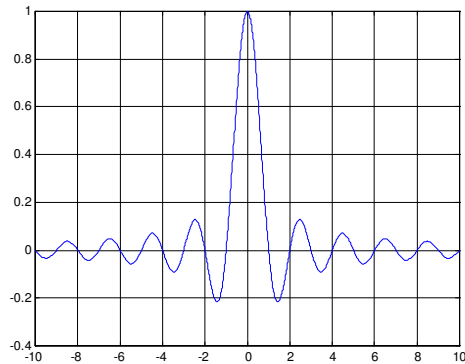
```
y=sinc(x);
```

```
plot(x,y)
```

```
grid
```

il software fornisce

```
%sinc(x) = sin(pi*x)/(pi*x)
```



Per la generazione di forme d'onda e segnali si veda nell'help **toolbox/signal**

Grafici multipli

`subplot (m, n, p)`

divide la finestra grafica in *m* righe e *n* colonne; *p* seleziona la finestra attiva del grafico *p* assume i valori da 1 a 4.

Esempio

```
%Grafica con MATLAB
```

```
t = [0:0.1:6];
```

```
y1=sin(2*3.14*1*t);
```

```
subplot(2, 2, 1);
```

```
plot(t,y1)
```

```
title('sen(t)')
```

```
ylabel('y(t)')
```

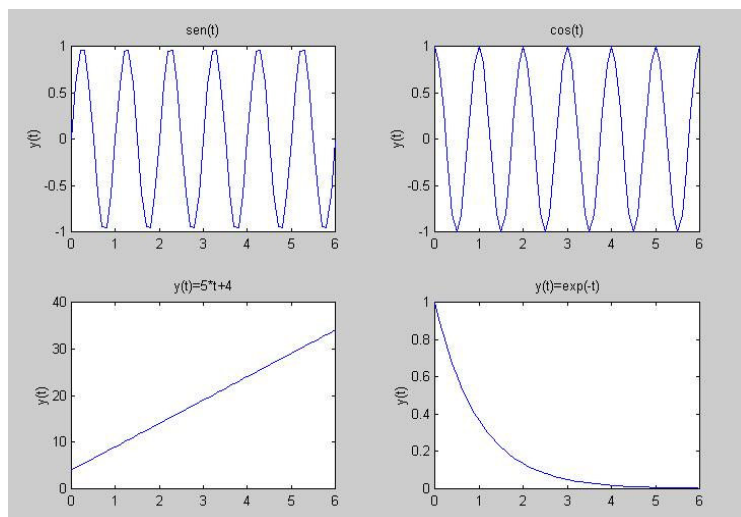
```
y2=cos(2*3.14*1*t);
```

```
subplot(2, 2, 2);
```

```
plot(t,y2)
```

```
title('cos(t)')
```

```
ylabel('y(t)')
```



```

y3=5*t+4;
subplot(2, 2, 3);
plot (t,y3)
ylabel('y(t)')
title('y(t)=5*t+4')

```

```

y4=exp(-t);
subplot(2, 2, 4);
plot(t,y4)
title('y(t)=exp(-t)')
ylabel('y(t)')

```

Funzioni di interesse elettronico

| | |
|----------|---|
| chirp | Swept-frequency cosine generator. |
| cos | Cosine of vector/matrix elements |
| diric | Dirichlet or periodic sinc function. |
| gauspuls | Gaussian-modulated sinusoidal pulse generator. |
| pulstran | Pulse train generator. |
| rectpuls | Sampled aperiodic rectangle generator. |
| sawtooth | Sawtooth or triangle wave generator. |
| sin | Sine of vector/matrix elements (see MATLAB Function Reference). |
| sinc | Sinc or $\text{sinc}(\pi t)/\pi t$ function. |
| tripuls | Sampled aperiodic triangle generator. |

Grafici tridimensionali

view (x, y, z)

specifica il punto di vista

Assegnata la funzione $z=f(x,y)$

per tracciare il grafico 3D si usa *mesh* o *surf*

mesh oppure surf

curve di livello nel piano x-y

_meshz oppure surfz

traccia una 'base'

cylinder(r)

Disegna la superficie di rotazione con generatrice descritta dal vettore r

Esempio

Sia $z = \sin x * \cos x$

per x compreso in [0, 4] per y compreso in [-2, 1]

$x = [0 : 0.1 : 4];$

$y = [-2 : 0.1 : 1];$

$[x, y] = \text{meshgrid}(x, y);$

$z = \sin(x) .* \cos(y);$

contour (x,y,z)

curve di livello

mesh (x,y,z)

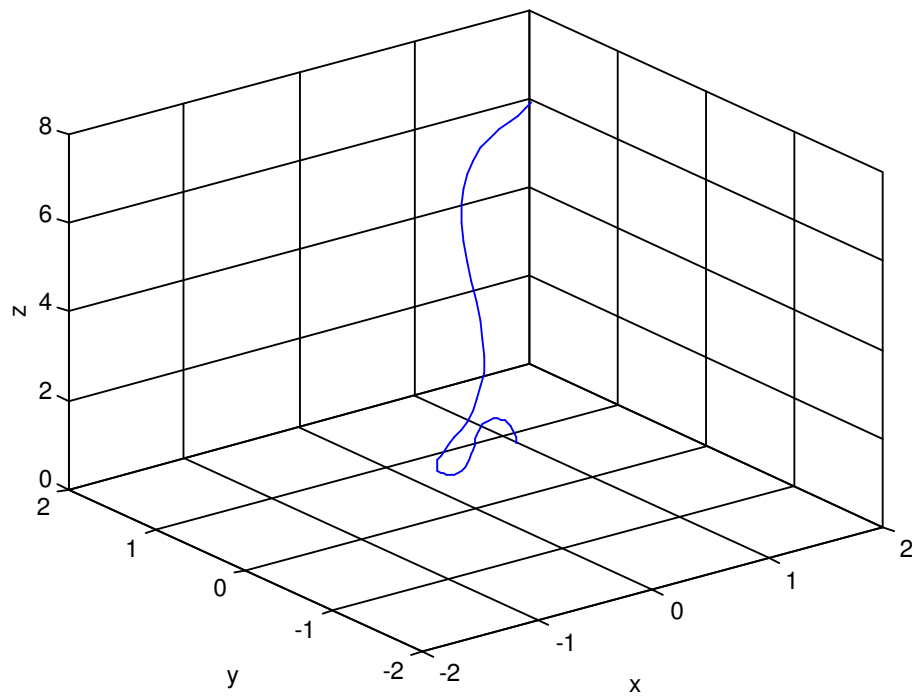
grafico 3D

Curve parametriche 3D

Esempio

Siano assegnati 3 vettori x ; y ; z

```
t = [0: 0.1: 2*pi];  
r = exp(t/10);  
x = r .* cos(t);  
y = r .* sin(x);  
z = t;  
plot3( x, y, z)  
grid;  
xlabel('x')  
ylabel('y')  
zlabel('z')
```



Control System Toolbox

E' l'insieme di funzioni per l'analisi e la simulazione dei controlli automatici.

Trasformata e Antitrasformata di Laplace

Esempio

Si riporta il listato di un programma in ambiente MATLAB 5 per il calcolo della trasformata di Laplace di una funzione assegnata. Nell'esempio $f(t) = a \cdot e^{-bt} + 12$.

disp('Calcolo della trasformata di Laplace della funzione f')

```
syms a b t s;           % Si definiscono le variabili per la trasformazione
f = a*exp(-b*t)+ 12    % f è la funzione che si desidera trasformare
F = laplace (f)         % F è la trasformata di Laplace della f
pretty(F)              % Visualizza in formato tipicamente matematico
```

Il software produce la seguente risposta

Calcolo della trasformata di Laplace della funzione f

f =
a*exp(-b*t)+12

F =
a/(s+b)+12/s

$$\frac{a}{s+b} + \frac{12}{s}$$

Esempio

Si riporta il listato di un programma in ambiente MATLAB 5 per il calcolo dell'antitrasformata di Laplace di una funzione assegnata.

Nell'esempio $F = a/(s^2 + a^2) - 6/(s+2)$.

```
Disp ('Calcolo dell"antitrasformata di Laplace della funzione F')
syms a,s,t;           % Si definiscono le variabili per la trasformazione
F = a/(s^2+a^2)-6/(s+2) % F è la funzione che si desidera antitrasformare
f = ilaplace (F)      % f è l'antitrasformata di Laplace della F
pretty(f)            % Visualizza in formato tipicamente matematico
```

Il software produce la seguente risposta

Calcolo dell'antitrasformata di Laplace della funzione F

F =
a/(s^2+a^2)-6/(s+2)

f =
sin(a*t)-6*exp(-2*t)
sin(at) - 6exp(-2t)

Funzione di trasferimento

Siano **n** e **d** i vettori dei coefficienti del numeratore e denominatore di $G(s) = N(s)/D(s)$

| | |
|---|--|
| <code>[zeri,poli] = tf2zp(n,d)</code> | Fornisce il valore degli zeri e dei poli |
| <code>[n,d] = zp2tf(zeri,poli)</code> | Fornisce N(s) e D(s) |
| <code>[residui, poli, Ao] = residue(n,d)</code> | Fornisce i residui, i poli e i termini diretti |
| <code>[n,d] = residue (residui, poli, Ao)</code> | Fornisce N(s) e D(s) |
| <code>fdt = tf(n,d)</code> | Fornisce la funzione di trasferimento |
| <code>fdt = zpk(zeri,poli,gain)</code> | Fornisce la funzione di trasferimento |

Esempio

```
z = [1,1];
p = [1,5,6];
K = 10
zpk(z,p,K)
```

fornisce
Zero/pole/gain:
 $10 (s-1)^2$

(s-1) (s-5) (s-6)

Esempio

```
h = tf([-10 20 0],[1 7 20 28 19 5])
```

fornisce
Transfer function:

$$(-10 s^2 + 20 s) / s^5 + 7 s^4 + 20 s^3 + 28 s^2 + 19 s + 5$$

```
» zpk(h)
fornisce
Zero/pole/gain:    -10 s (s-2) / (s+1)^3 (s^2 + 4s + 5)
```

Esempio

```
[z,p] = tf2zp([1,2],[1,5,6])
fornisce
```

```
z =
-2
p =
-3
-2
```

Funzioni per il Controllo Automatico

| | |
|---|---|
| [n, d] = series (n1, d1, n2, d2) | calcola la funzione di trasferimento di due blocchi in cascata: $n = n1*n2$ e $d = d1*d2$. |
| [n, d] = feedback (n1,d1, n2,d2,, sign) | calcola la funzione di trasferimento $W(s)$ di due blocchi in retroazione. Il blocco diretto $G(s) = n1/d1$ e quello di retroazione $H(s) = n2/d2$. Il parametro sign se omesso indica reazione negativa. |
| [n, d] = cloop (n1, d1, sign) | calcola la funzione di trasferimento ad anello chiuso per un sistema a retroazione unitaria $H(s) = 1$ e $G(s) = n1/d1$. |
| [y, x, t] = step (n, d, t) | calcola la risposta al gradino della $G(s) = n/d$. y è il vettore che contiene i valori della risposta negli istanti t . Il vettore x contiene la risposta nello spazio degli stati. |
| [y, x, t] = impulse (n, d, t) | risposta alla Delta di Dirac |
| [y, x, t] = Isim (n, d, t) | risposta ad un segnale definito dall'utente |
| [wn, zita] = damp(d) | pulsazione naturale e smorzamento |
| dcgain(n,d) | guadagno statico |
| [p,z] = pzmap(n,d) | mapa grafica poli (con x) e zeri con (o) |

Esempio

Risposta al gradino di un sistema a retroazione unitaria con $G(s) = N(s)/D(s)$.
E' noto $N(s) = s + 4$ e $D(s) = s^2 + 5s + 6$

```
t = [0: 0.1: 20]; n = [1, 4]; d = [ 1, 5, 6];
[num, den] = cloop (n, d);
[y, x, t] = step (num, den, t) ;
plot ( t,y)
xlabel (' tempo in s')
ylabel( ' U(t)')
title (' Risposta nel dominio del tempo ')
```

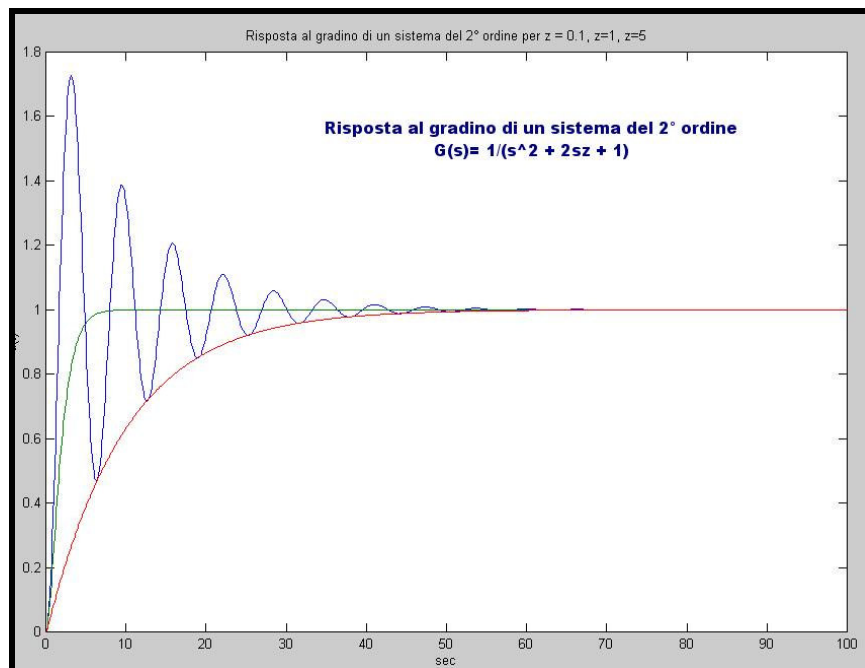
Esempio

Risposta al segnale a rampa unitaria

```
t = [0: 0.1: 20]; n = [1, 4]; d = [ 1, 5, 6];
[num, den] = cloop (n, d);
% Definizione rampa
u = t
[y, x, t] = Isim (num, den, u, t) ;
plot ( t,y, t, u)
xlabel ( ' tempo in s' )
ylabel( ' Uscita ' )
title ( ' Risposta nel dominio del tempo ' )
```

Esempio

```
disp('Risposta al gradino di un sistema del 2° ordine')
disp(' G(s)= 1/(s^2 + 2sz + 1) per diversi valori di z')
t= [0:0.1:100]; num=[1];
z1= 0.1; den1=[1, 2*z1,1];
z2= 1; den2=[1, 2*z2,1];
z3= 5; den3=[1, 2*z3,1];
[y1,x,t]=step(num,den1,t);
[y2,x,t]=step(num,den2,t);
[y3,x,t]=step(num,den3,t);
plot(t,y1,t,y2,t,y3)
title('Risposta al gradino di un sistema del 2° ordine per z = 0.1, z=1, z=5')
xlabel('sec')
ylabel('u(t)')
```



Diagrammi di Bode Nyquist, Evans, Nichols

[mag, phase, w] = bode (n, d, w) costruisce i vettori modulo (mag) e fase (phase)
 [mag, phase, w] = bode (n, d,) traccia il diagramma di Bode

[Gm, Pm, wcg, wcp] = margin (mag, phase, w)

dove: Gm è il margine di ampiezza in corrispondenza della pulsazione wcg e Pm è il margine di fase in corrispondenza di wcp.

Bode (n,d) traccia il diagramma di Bode
 margin (mag, phase, w) traccia il diagramma di Bode con i relativi margini.
 Nyquist (n, d) traccia il luogo di Nyquist
 rlocus (n, d) traccia il luogo delle radici
 nichols (n, d) traccia il diagramma di Nichols

Si è indicato con n e d i vettori contenenti i coefficienti del numeratore e denominatore della G(s).

Esempio

f = zpk([], [-2, -3, -4], 20); calcola $G(s) = 20 / (s+2)(s+3)(s+4)$
 bode (f) traccia il diagramma di Bode
 nyquist (f) traccia il diagramma di Nyquist

Esempio

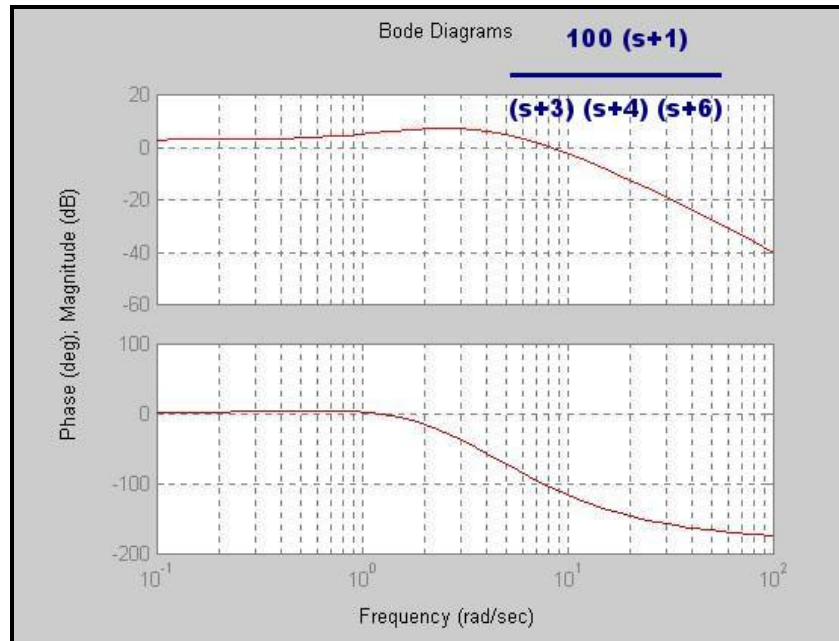
n = [200]; d = [1, 15, 120] %G(s) = $200/(s^2 + 15s + 120)$
 %Calcolo dei poli
 r = roots (d)
 %Antitrasformata
 ilaplace ('200/ (s^2 + 15*s +120)')
 pause
 [Modulo, Fase, w] = bode (n, d);
 semilogx (w/(2*pi), 20*log10 (Modulo);
 xlabel (' Frequenza in Hz');
 ylabel ('Modulo in dB');
 grid;

Esempio

% tracciato di Bode
 disp('Diagramma di Bode ')

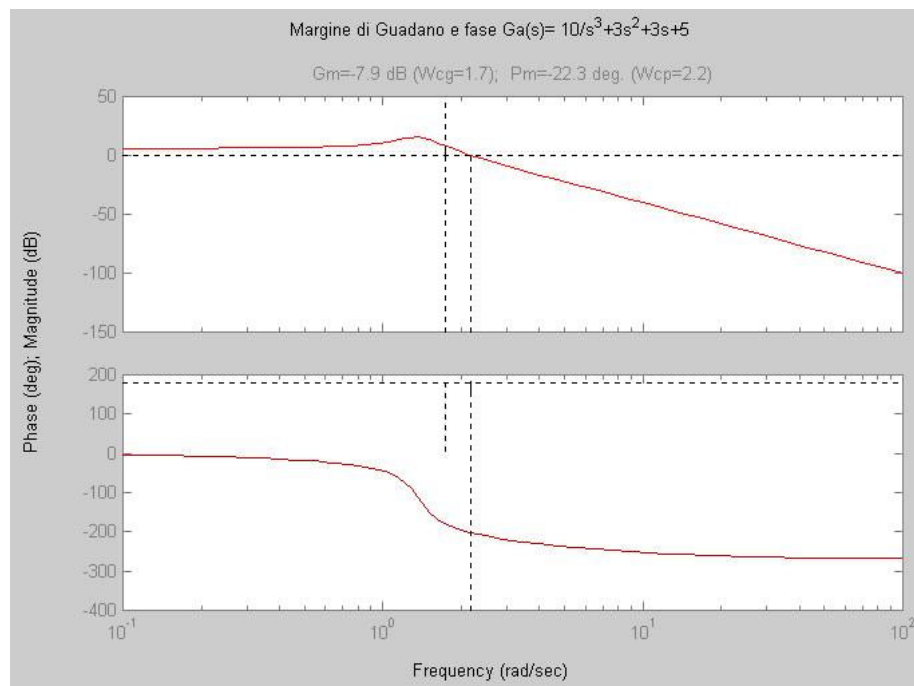
f = zpk([-1],[-3,-4,-6],100) %G(s) = $100(s+1)/[(s+3)(s+4)(s+6)]$
 bode(f)

Si ottiene



Esempio: Margine di Guadagno e di fase

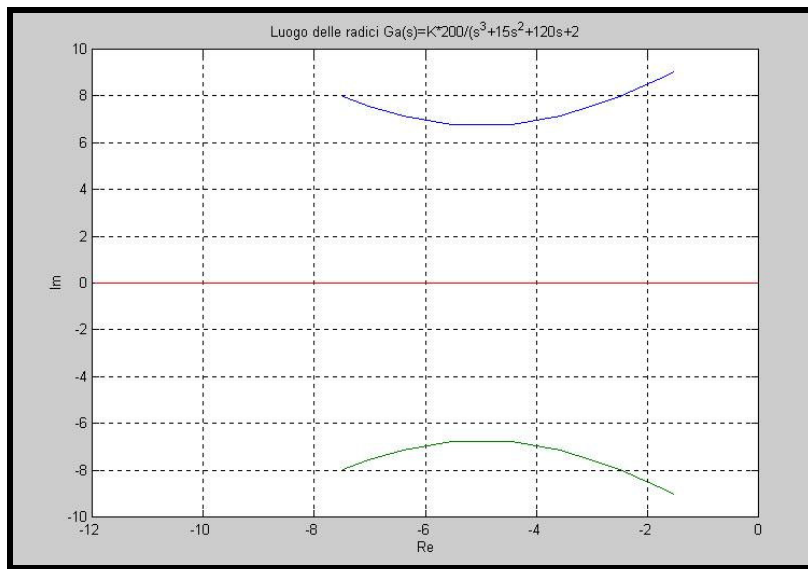
```
disp('Margine di guadagno e fase di Ga(s)= 10/s^3+3s^2+3s+5')
margin( 10,[1,3,3,5])
title('Margine di guadagno e fase Ga(s)= 10/s^3+3s^2+3s+5')
```



Esempio

Luogo delle radici

```
disp('Luogo delle radici della Ga(s)=K*200/(s^3+15s^2+120s+2)')
K=0:0.5:5;
n=[200];
d=[1,15,120,2];
[r K]=rlocus(n,d,K);
plot(r, '-')
grid;
title('Luogo delle radici Ga(s)=K*200/(s^3+15s^2+120s+2)');
xlabel('Re');
ylabel('Im');
```



TRASFORMATA Z

Trasformata z

```
syms k n w z
ztrans(2^n)      fornisce. z/(z-2)
ztrans(sin(k*n),w)  fornisce: sin(k)*w/(1-2*w*cos(k)+w^2)
```

Antitrasformata z

```
iztrans(z/(z-2))      fornisce: 2^n.
iztrans(exp(x/z),z,k)  fornisce: x^k/k!.
```

TRASFORMAZIONI CONTINUO - DISCRETO E DISCRETO - CONTINUO

Si descrivo le funzioni fondamentali per la conversione di unzioni dal continuo (dominio della trasformata di Laplace) al discreto (dominio della trasformata z)

```
c2dm      Conversione da continuo a discreto con metodo
d2cm      Conversione discreto continuo con metodo
```

I metodi utilizzati sono:

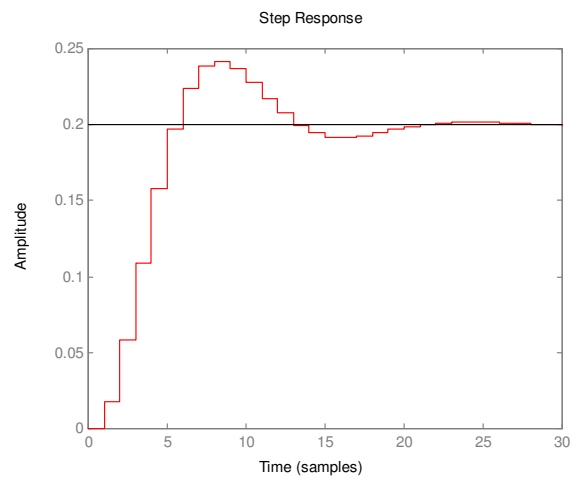
| | |
|-----------|--|
| 'zoh' | Zero-order hold. The control inputs are assumed piecewise constant over the sampling period T_s . |
| 'foh' | Triangle approximation (modified first-order hold). The control inputs are assumed piecewise linear over the sampling period T_s . |
| 'tustin' | Bilinear (Tustin) approximation. |
| 'prewarp' | Tustin approximation with frequency prewarping. |
| 'matched' | Matched pole-zero method. |

Esempio

```
disp('Trasformata z')
disp('Risposta al gradino di un sistema del secondo ordine tempo-
discreto')
disp('sia G(s) = 1/(s^2+2*s +5) il sistema del secondo ordine tempo
continuo')
%Nel continuo G(s)
num=[1];den=[1,2,5];
%Calcolo poli della G(s)
roots(den)
%Nel tempo discreto G(z)
[numd,dend]=c2dm(num,den,0.2,'zoh')
%Risposta al gradino nel tempo discreto
dstep(numd,dend)
```

Il software produce la seguente risposta

```
ans =
-1.0000+ 2.0000i
-1.0000- 2.0000i
numd =
0 0.0173 0.0151
dend =
1.0000 -1.5082 0.6703
```



Diagrammi di Bode e Nyquist nel dominio tempo - discreto

Sono equivalenti al tempo continuo con gli stessi comandi

Esempio

disp('Diagramma di Bode di un sistema del secondo ordine tempo-discreto')
 disp('sia $G(s) = 1/(s^2+2*s +5)$ il sistema del secondo ordine tempo continuo')

%Nel continuo G(s)

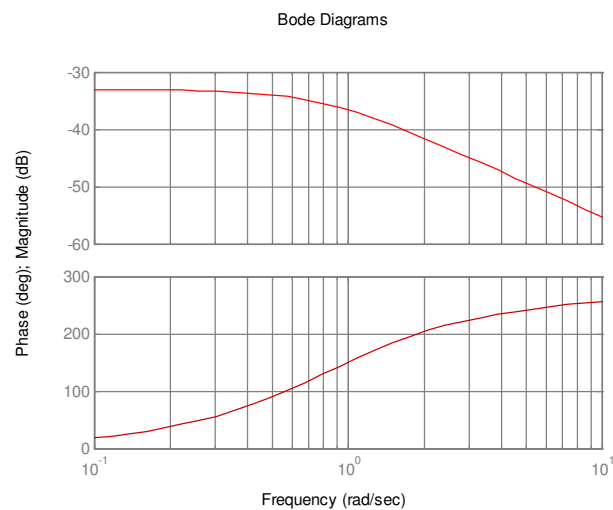
num=[1];den=[1,2,5];

[numd,dend]=c2dm(num,den,0.2,'zoh')

%Diagramma di Bode nel tempo discreto

bode(numd,dend)

Si ottiene



Programmazione con MATLAB

E' possibile programmare utilizzando un'insieme di istruzioni. Le fondamentali sono:

Selezione

```

if condizione
else if condizione
istruzione
"
else
"
"
end

```

```

switch espressione
case .....
case.....
otherwise ....
end

```

Iterazione

```

for i=1 : n
"
"
end

```

oppure

```
T = [0: 0.1: 2]
```

```

for t =T
"
"
end

```

oppure

```
while ( condizione)
```

```

"
"
end

```

Altre

```

Pause ( n )      sospende per n secondi
n = input ( '.....' ) istruzione di input
%-----        commento
disp ( 'commento...' ) mostra il commento sul video

```

SIMULINK

Simulink è un software di simulazione integrato nell'ambiente MATLAB.

Simulink dispone di una vasta biblioteca di blocchi funzionali di facile e immediato utilizzo.

I blocchi si richiamano e si gestiscono con le tipiche funzioni del mouse in ambiente Windows:

- **Tasto sinistro.** Seleziona e sposta un blocco. Disegna linee di collegamento. Dopo la selezione il blocco si cancella premendo CANC.
- **Tasto destro:** Duplica un blocco. Consente la diramazione e il disegno di linee.
- **Doppio click:** Gestisce le proprietà del blocco.
- **Doppio tasto:** Selezione multipla. Analogo al tasto sinistro + Shift. Disegna una linea che si sposta con continuità. Consente di muovere i blocchi con continuità.

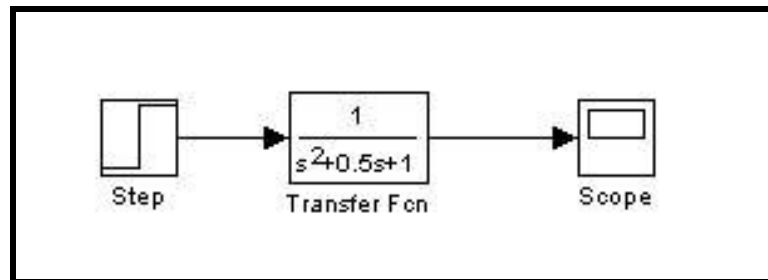
Ogni blocco è caratterizzato da due proprietà:

Stilistiche (colore, font, orientazione,..)

Strutturali (nome, parametri,...)

Esempio

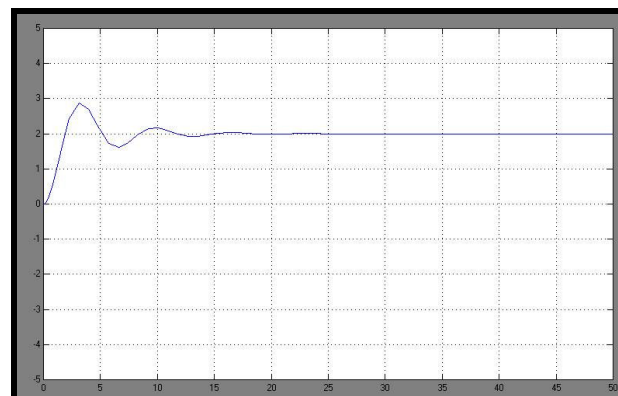
Costruire in ambiente simulink lo schema seguente.



Con doppio click sul generatore si impostano i parametri del generatore:

Step time =0, initial value = 0 final value = 2

Con doppio click su Scope si apre l'oscilloscopio. Attivare dal menu Simulation la funzione START. Modificare, se necessario i parametri della simulazione dal menu Simulation/Parameters . Regolare la scala dell'oscilloscopio per ottimizzare la lettura. Si ottiene:



Per interrompere la simulazione attivare il menu *Simulation/Stop*.

Per fissare a priori la durata della simulazione attivare il sottomenu *Parameters* ed assegnare i valori a *start time* e *stop time*.

GROUP

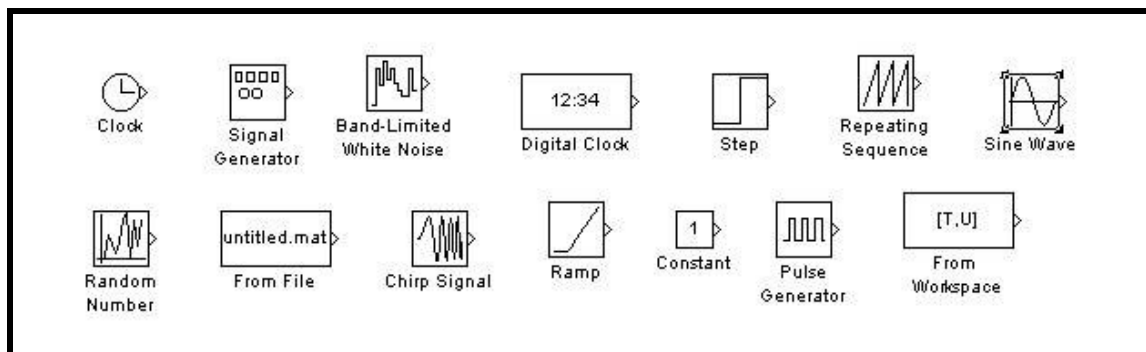
Uno schema a blocchi complesso può essere raggruppato in un solo blocco attivando la funzione *create subsystem* dal menu *edit*. Prima di attivare la funzione di raggruppamento si devono selezionare i blocchi da raggruppare. Per analizzare la struttura interna del blocco creato è sufficiente il solito doppio click. Compare lo schema con l'aggiunta di nuovo blocchi d'ingresso e di uscita. E' possibile raggruppare più blocchi per ottenere un superblocco.

LIBRERIA DEI BLOCCHI

Si riportano i blocchi funzionali più importanti rimandando alla guida in linea e alle demo l'approfondimento dell'argomento.

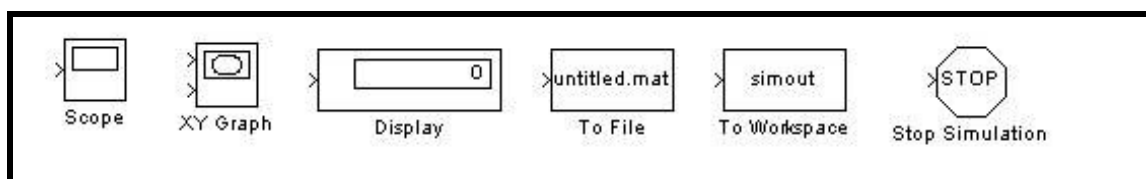
SOURCE

Blocchi per la generazione dei segnali di ingresso.



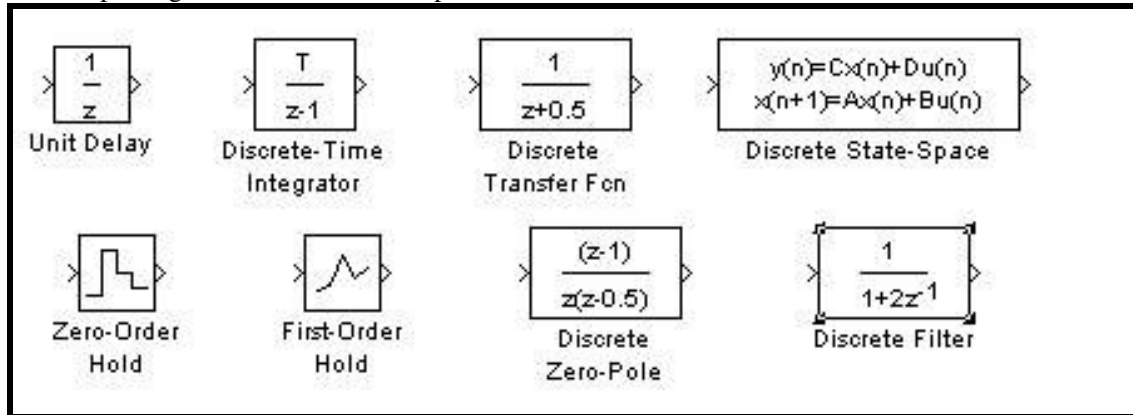
SINK

Blocchi per l'analisi dei dati di uscita.



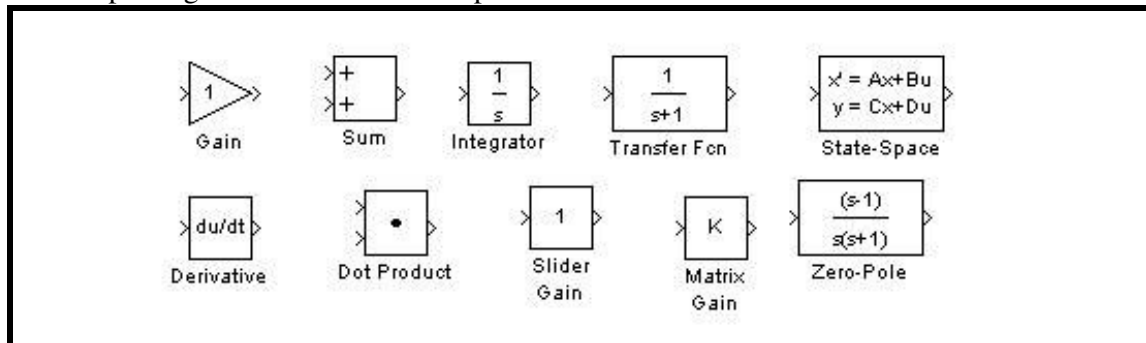
DISCRETE

Blocchi per la gestione dei sistemi tempo discreti



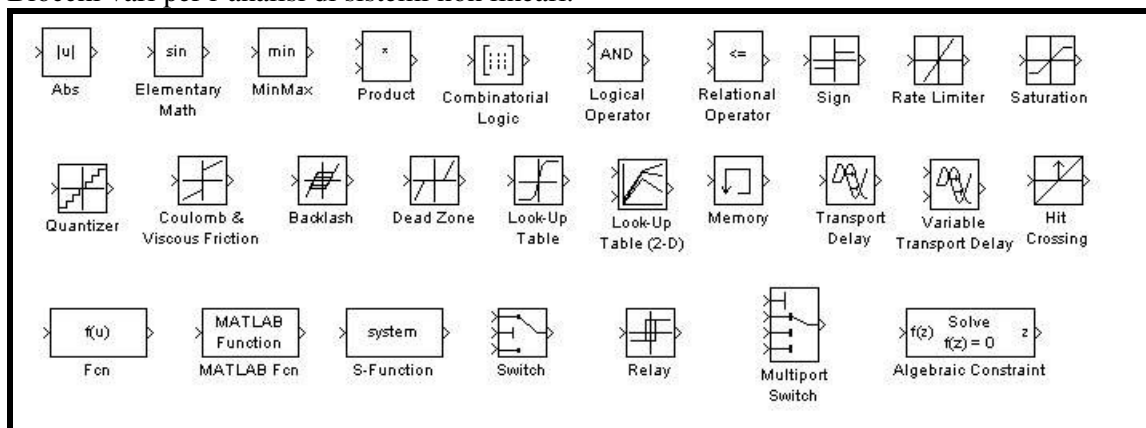
LINEAR

Blocchi per la gestione dei sistemi tempo continui.



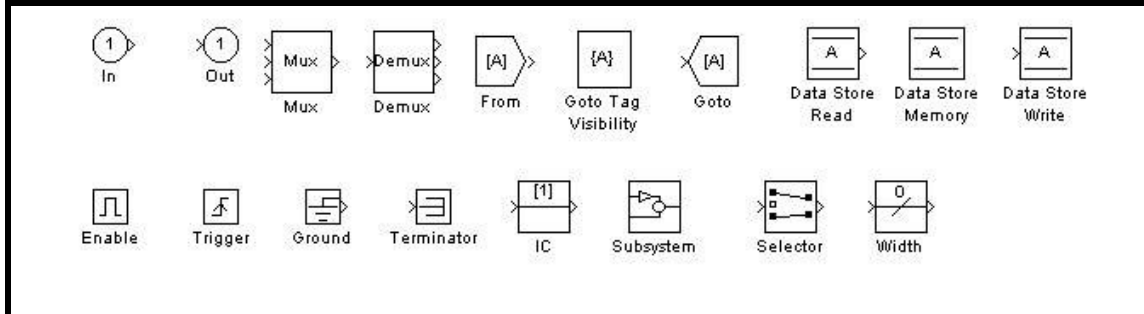
NONLINEAR

Blocchi vari per l'analisi di sistemi non lineari.



CONNECTION

Blocchi vari.

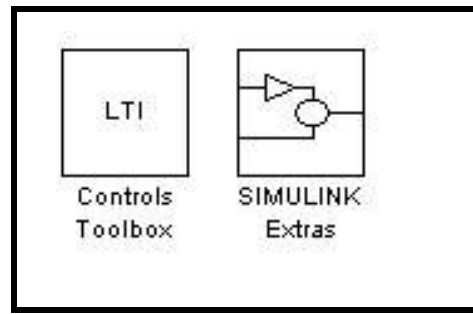


BODSET & TOOLBOXES

Doppio click per aprire i sottoblocchi interni.

LTI: Blocco per la gestione dei sistemi Lineari Tempo Invariabili.

Extras. Ulteriore insieme di blocchi funzionali sia di generatori che di sistemi analogici e digitali.



Si consiglia di analizzare le interessanti demo presenti nel pacchetto.

ESEMPI VARI

Esempio

Analisi di un amplificatore ad emettitore comune con Matlab

Programma

```

disp( 'Inserimento dei dati')

IC = input('Inserire il valore di IC in mA:      IC= ');
VCE = input('Inserire il valore di VCE in Volt:   VCE = ');
VCC = input('Inserire il valore di VCC in Volt:   VCC = ');
SI = input('Inserire il valore di SI:            SI = ');
Rs = input('Inserire il valore di RS in Kohm:     RS = ');
VBE = input('Inserire il valore di VBE in Volt:   VBE = ');
hFE = input('Inserire il valore di hFE= hfe:     hFE = hfe = ');
hie = input('Inserire il valore di hie in Kohm:   hie = ');
hoe = input('Inserire il valore di hoe in µA/V:   hoe= ');
VsM = input('Inserire il valore di VsM in mV:     VsM= ');

disp('Analisi in continua')
disp('Le resistenze sono in KOhm, le tensioni in Volt e le correnti in mA')

VRE = 0.15*VCC
RE = VRE/IC
RC=(VCC-VCE-VRE)/IC
IB=IC/hFE
RB=(SI-1)*RE
VBB = RB*IB +VBE+VRE
R1=VCC*RB/VBB
R2=R1*RB/(R1-RB)

disp('Analisi in alternata')

Ai=hFE/(1+0.001*hoe*RC)
Av=-1*Ai*RC/hie
Ro=10^3/hoe
Rg=RB*hie/(RB+hie)
a=Rg/(Rg+Rs)
Avs=a*Av
ViM=a*VsM

disp('Grafica')
t=[0:0.05:2];
Vs=VsM*sin(2*3.14*t);
subplot(2,1,1);
plot(t,Vs)
grid;
title('Tensione di entrata Vs(t)')
ylabel('Tensione in mV')
Vo= Avs*VsM*sin(2*3.14*t);
subplot(2,1,2);
plot(t,Vo)
grid;
title('Tensione di uscita Vo(t)')
xlabel('Tempo in secondi')
ylabel('Tensione in mV')

```

Il software produce il seguente output

Inserimento dei dati

Inserire il valore di IC in mA: IC= 2
Inserire il valore di VCE in Volt: VCE = 5
Inserire il valore di VCC in Volt: VCC = 9
Inserire il valore di SI: SI = 10
Inserire il valore di RS in Kohm: RS = 1
Inserire il valore di VBE in Volt: VBE = 0.65
Inserire il valore di hFE= hfe: hFE = hfe = 200
Inserire il valore di hie in Kohm: hie = 2.5
Inserire il valore di hoe in $\mu\text{A/V}$: hoe= 25
Inserire il valore di VsM in mV: VsM= 10

Analisi in continua

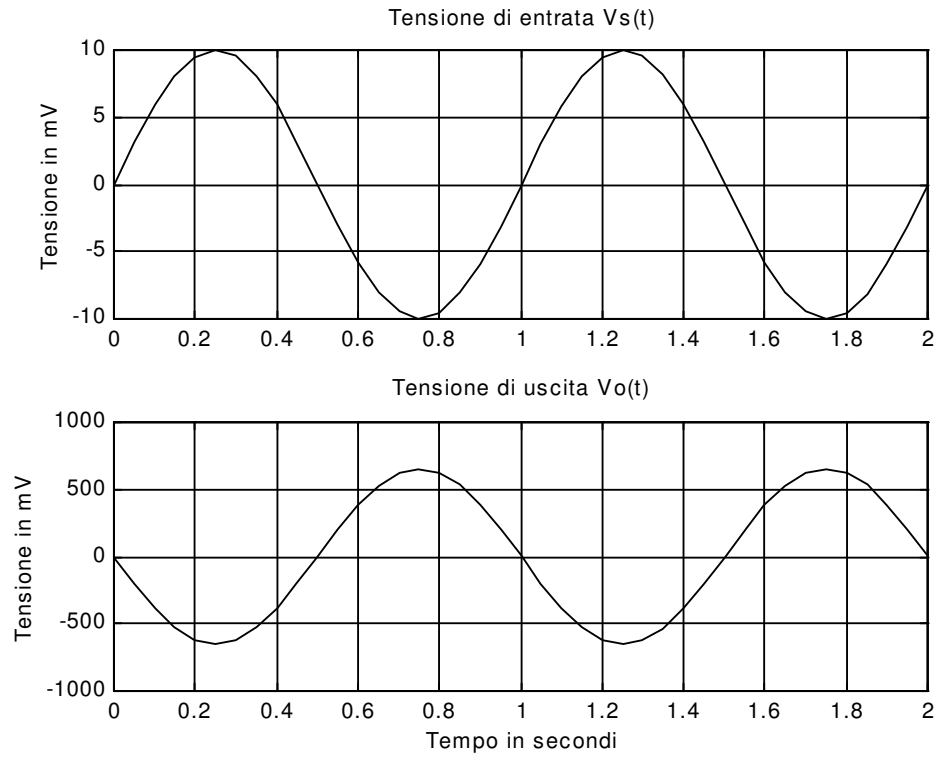
Le resistenze sono in KOhm, le tensioni in Volt e le correnti in mA

VRE =
1.3500
RE =
0.6750
RC =
1.3250
IB =
0.0100
RB =
6.0750
VBB =
2.0607
R1 =
26.5316

R2 =
7.8791

Analisi in alternata

Ai =
193.5874
Av =
-102.6013
Ro =
40
Rg =
1.7711
a =
0.6391
Avs =
-65.5763
ViM =
6.3914

Grafica**Esempio**

Si riporta il listato in MATLAB 5 per il calcolo dei poli e degli zeri note le funzioni N(s) e D(s).

```
disp('Calcolo: Zeri, Poli noto N(s) e D(s)')
num = [4, 8, 0];           %Numeratore   num = 4*s^2 + 8*s
den = [1, 4, 3, 5];       %Denominatore den = s^3 + 4*s^2 + 3s + 5
[Zeri, Poli]=tf2zp(num,den)
```

Il software produce la seguente risposta

Calcolo: Zeri, Poli noto N(s) e D(s)

```
Zeri =
    0
   -2
Poli =
  -3.5517
 -0.2241+ 1.1651i
 -0.2241- 1.1651i
```

Esempio

Si riporta il listato di un programma in MATLAB 5 per il calcolo dei residui, dei poli e dei termini diretti di una funzione di trasferimento G(s) note le funzioni N(s) e D(s).

```
disp('Calcolo dei residui')
num = [4, 8, 0]; %Numeratore N(s) = 4*s^2 + 8*s
den = [1, 4, 3]; %Denominatore D(s) = s^2 + 4*s + 3
[residui, poli, Ao]=residue(num, den)
```

Il software produce la seguente risposta

Calcolo dei residui

residui =

-6

-2

poli =

-3

-1

Ao =

4

Esempio

Si riporta il listato di un programma in MATLAB 5 per il calcolo di N(s) e D(s) noti i residui, i poli e i termini diretti.

```
disp('Calcolo di N(s) e D(s) noti i residui, i poli e i termini diretti')
residui=[-6,-2];
poli=[-3,-1];
Ao=[4];
[numeratore,denominatore]=residue(residui,poli,Ao)
```

Il software produce la seguente risposta

Calcolo di N(s) e D(s) noti i residui, i poli e i termini diretti

numeratore =

4 8 0

denominatore =

1 4 3

Esempio

Si riporta il listato di un programma scritto in ambiente MATLAB 5 per tracciamento dei diagrammi di Bode o Nyquist di sistemi del 1° ordine.

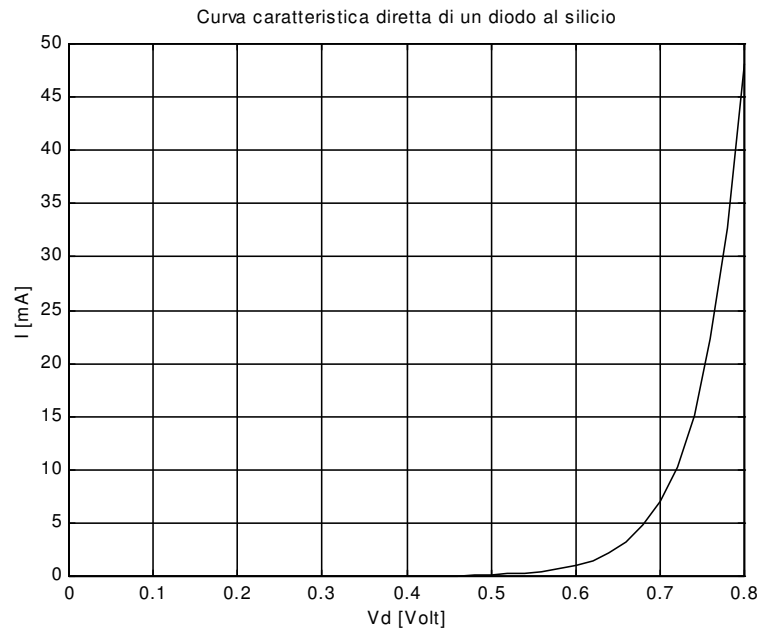
```
disp('Diagrammi di Bode e Nyquist di alcune funzioni del primo ordine ')
disp('')
disp(' 1 - Bode - Filtro passa basso K/(s + w)')
disp(' 2 - Bode - Filtro passa alto Ks/(s + w)')
disp(' 3 - Bode - Rete corretttrice k(s + a)/(s + b)')
disp(' 4 - Nyquist - Filtro passa basso K/(s + w)')
disp(' 5 - Nyquist - Filtro passa alto Ks/(s + w)')
disp(' 6 - Nyquist - Rete corretttrice k(s + a)/(s + b)')
disp('')
n=input('Quale digramma vuoi tracciare? = ');
switch n
case 1
    K = input(' Immetti il valore di K = ');
    w=input(' Immetti il valore di w = ');
    num=[K]; den=[1, w];
    bode(num,den)
    xlabel('w (rad/sec)')
    title('Filtro passa - basso')
    grid;
case 2
    K = input(' Immetti il valore di K = ');
    w=input(' Immetti il valore di w = ');
    num=[K, 0]; den=[1, w];
    bode(num,den)
    xlabel('w (rad/sec)')
    title('Filtro passa - alto')
    grid;
```

```
case 3
    K = input(' Immetti il valore di K = ');
    a =input('  Immetti il valore di a = ');
    b =input('  Immetti il valore di b = ');
    num=[K, K*a]; den=[1, b];
    bode(num,den)
    xlabel('w (rad/sec)')
    title('Rete corretrice')
    grid;
case 4
    K = input(' Immetti il valore di K = ');
    w=input('  Immetti il valore di w = ');
    num=[K]; den=[1, w];
    nyquist(num,den)
    title('Filtro passa - basso')
    grid;
case 5
    K = input(' Immetti il valore di K = ');
    w=input('  Immetti il valore di w = ');
    num=[K, 0]; den=[1, w];
    nyquist(num,den)
    xlabel('w (rad/sec)')
    title('Filtro passa - alto')
    grid;
case 6
    K = input(' Immetti il valore di K = ');
    a =input('  Immetti il valore di a = ');
    b =input('  Immetti il valore di b = ');
    num=[K, K*a]; den=[1, b];
    nyquist(num,den)
    title('Rete corretrice')
    grid;
otherwise
    disp(' La scelta è errata')
end
```

Esempio

Curva caratteristica di un diodo

```
disp('Curva caratteristica di un diodo al silicio')
Vd = [0:0.02:0.8];
I0=10^-8;
VT=0.025;
a=2;
V=a*VT;
I=1000*I0*(exp(Vd/V)-1);
plot(Vd,I)
grid;
xlabel('Vd [Volt]')
ylabel('I [mA]')
title('Curva caratteristica diretta di un diodo al silicio')
```



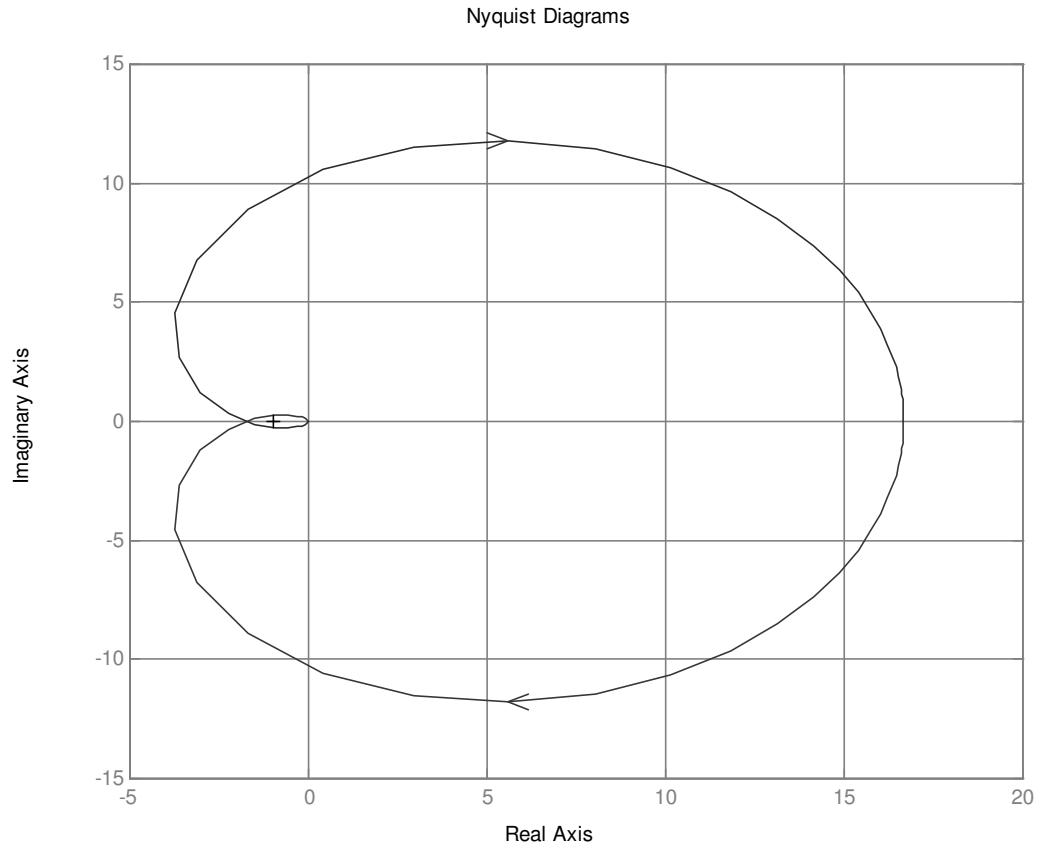
Esempio

Si riporta il listato di un programma in ambiente MATLAB per il tracciamento del diagramma di Nyquist della funzione $G(s) = 500 / (s^2 + 5s + 6) (s + 5)$

Programma

```
disp ('Diagramma di Nyquist della G(s) = 500/(s^2+5s+6) (s+5)')
numeratore = [500]; d1=[1,5,6];d2=[1,5];
denominatore = conv(d1,d2);
%Diagramma di Nyquist per  $\omega$  variabile tra  $-\infty$  e  $+\infty$ 
nyquist(numeratore, denominatore)
grid;
```

Il software produce la seguente diagramma



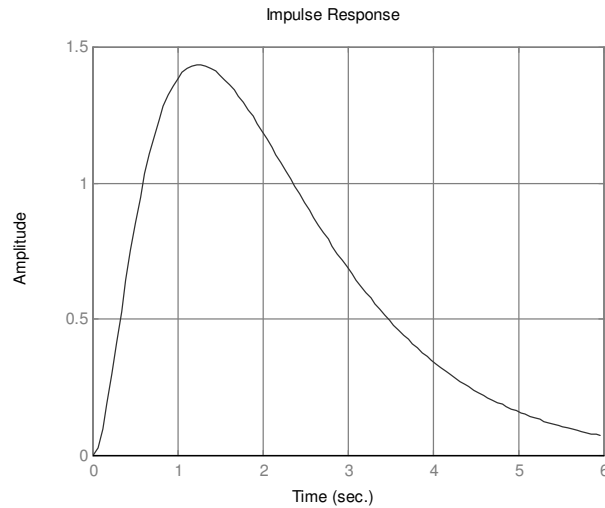
Esempio

Si riporta il listato di un programma in ambiente MATLAB per il calcolo dei poli e della risposta alla Delta di Dirac della $G(s) = 20/[(s^2+2s+1)(s+5)]$.

```
disp ('Determinare i poli della G(s) = 20/[(s^2+2s+1)(s+5)] e la risposta
alla delta di Dirac')
numeratore=[20];
d1=[1,2,1];d2=[1,5];
denominatore=conv(d1,d2)           %Calcola il prodotto tra d1 e d2
poli = roots(denominatore)         %Calcola i poli
impulse (numeratore , denominatore) %Grafico della risposta alla Delta di
Dirac
grid;                               %Inserisce la griglia nel grafico
```

Il software produce la seguente risposta

```
denominatore =
    1    7   11    5
poli =
   -5.0000
  -1.0000+ 0.0000i
  -1.0000- 0.0000i
```



Esempio

Si riporta il listato di un programma in ambiente MATLAB per lo studio di un sistema a reazione negativa noti i valori delle funzioni di trasferimento $G(s)$ del blocco diretto e $H(s)$ del blocco di reazione. Il programma calcola la funzione di trasferimento totale con reazione $W(s)$ e i relativi poli e zeri. Inoltre si valuta anche il guadagno d'anello $Ga(s)$ e si disegna il corrispondente diagramma di Bode e di Nyquist. Nell'esempio: $G(s) = 45/(s^3 + 2s^2)$ e $H(s) = 0.2$

```
disp('Analisi di una rete a reazione negativa')
%Numeratore e denominatore della funzione G(s)
NUMG=[45]; DENG=[1,2,0,0];
%Numeratore e denominatore della funzione H(s)
NUMH=[0.2]; DENH=[1];
%Calcolo della funzione con reazione W(s)= G(s)/[1+G(s)H(s)]
[NUMW,DENW]= feedback(NUMG,DENG,NUMH,DENH)
%Calcolo del guadagno d'anello Ga(s) = G(s)H(s)
NUMGA= conv(NUMG,NUMH)
DENGA = conv(DENG,DENH)
%Calcolo degli zeri e dei poli della funzione W(s)
[ZERIW,POLIW]=tf2zp(NUMW,DENW)
disp('Diagramma di Bode e di Nyquist del guadagno di anello')
figure (1)
bode(NUMGA,DENGA)
xlabel(' Pulsazione in rad/s')
figure (2)
nyquist(NUMGA,DENGA)
grid;
```

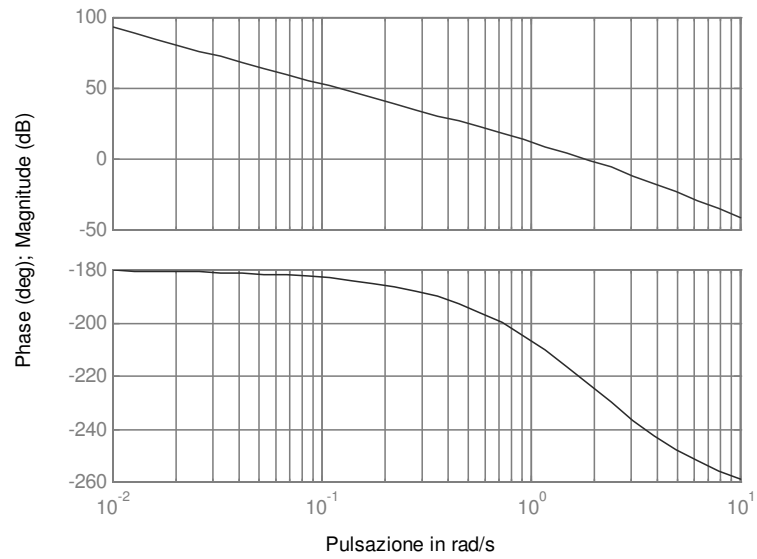
Il software produce la seguente risposta

```
Analisi di una rete a reazione negativa
NUMW =
    0    0    0   45
DENW =
    1    2    0    9
NUMGA =
     9
DENGA =
```

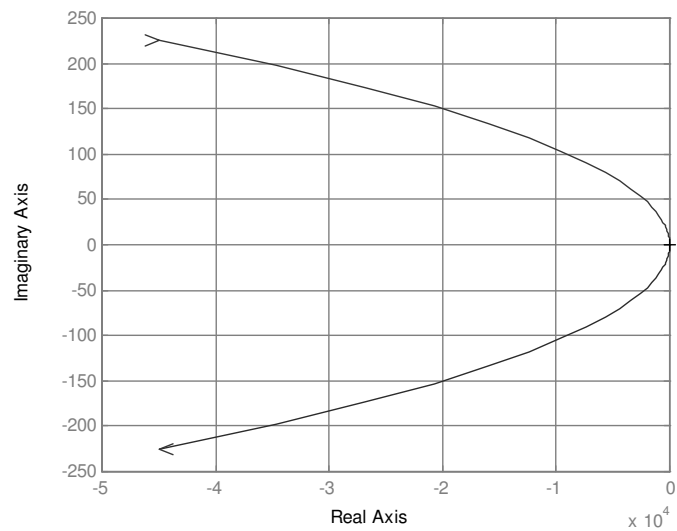
```
1 2 0 0
ZERIW =
Empty matrix: 0-by-1
POLIW =
-3.0000
0.5000+ 1.6583i
0.5000- 1.6583i
```

Diagramma di Bode e di Nyquist del guadagno di anello

Bode Diagrams

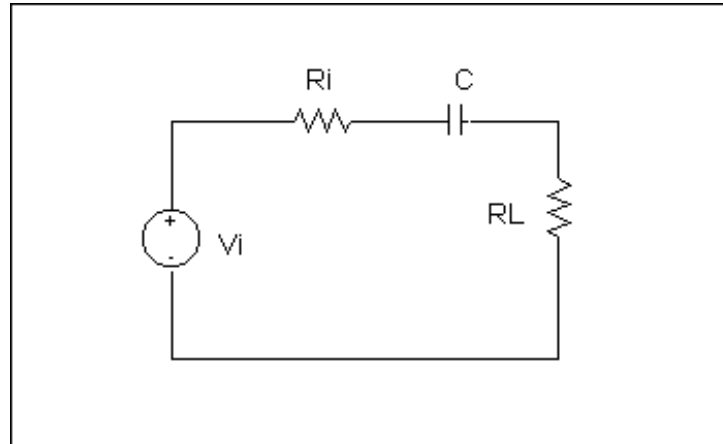


Nyquist Diagrams



Esempio

Si riporta il listato di un programma in ambiente MATLAB 5 per la risoluzione di della seguente rete RC.



```

clear;
disp('Rete elettrica del 1° ordine a componenti RC fig. 26 Cap.8')
disp('Dati di ingresso del sistema:')
disp('')
Ri=input('Resistenza di ingresso Ri in [Ohm] = ');
RL=input('Resistenza di carico RL in [Ohm] = ');
C=input('Capacità C in [Farad] = ');
A=input('Ampiezza ViM del segnale di ingresso in [Volt] = ');
w= input('Pusazione w in [rad/s]=');
tau=(Ri+RL)*C;
disp('')
disp(sprintf('La costante di tempo tau[μs] = %0.2f',tau*10^6));
wn=1/tau;
ft=wn/(2*pi)
disp('')
disp(sprintf('La frequenza di taglio ft[KHz]= %0.2f',ft/10^3));
td=0.35/ft;
disp(sprintf('Il tempo di discesa td[μs]= %0.2f',td*10^6));
disp('Calcolo del numeratore e del denominatore della G(s)')
K=RL/(Ri+RL);
numeratore=[K,0]
denominatore=[1,wn]
disp('Calcolo zeri e poli')
z=roots(numeratore);
p=roots(denominatore);
disp(sprintf('La funzione G(s) è caratterizzata da uno zero nullo
z=%0.1f',0));
disp(sprintf('La funzione G(s)è caratterizzata da un polo p= %0.2f',-
wn));
%Calcolo del modulo e della fase della funzione di trasferimento
modulo = K/sqrt(1+(wn/w)^2)
fase=atan(wn/w)
fasegradi= fase*180/pi
disp('Calcolo del valore di picco VoM della tensione di uscita')
VoM=modulo*A
disp('Risposta al gradino unitario')
disp('Premi un tasto per visualizzare il grafico della risposta al
gradino `)

```

```

pause;
step( numeratore, denominatore)    %Grafico della risposta al gradino
grid;

```

Il software produce la seguente risposta

Rete elettrica del 1° ordine a componenti RC
 Dati di ingresso del sistema:

Resistenza di ingresso R_i in [Ohm] = 500
 Resistenza di carico R_L in [Ohm] = $2 \cdot 10^3$
 Capacità C in [Farad] = $10 \cdot 10^{-9}$
 Ampiezza V_{iM} del segnale di ingresso in [Volt] = 5
 Pusazione w in [rad/s] = 10^4

La costante di tempo τ [μ s] = 25.00
 La frequenza di taglio f_t [KHz] = 6.37
 Il tempo di discesa t_d [μ s] = 54.98

Calcolo del numeratore e del denominatore della $G(s)$

numeratore =
 0.8000 0

denominatore =
 1 40000

Calcolo zeri e poli

La funzione $G(s)$ è caratterizzata da uno zero nullo $z = 0.0$

La funzione $G(s)$ è caratterizzata da un polo $p = -40000.00$

modulo =
 0.1940

fase =
 1.3258

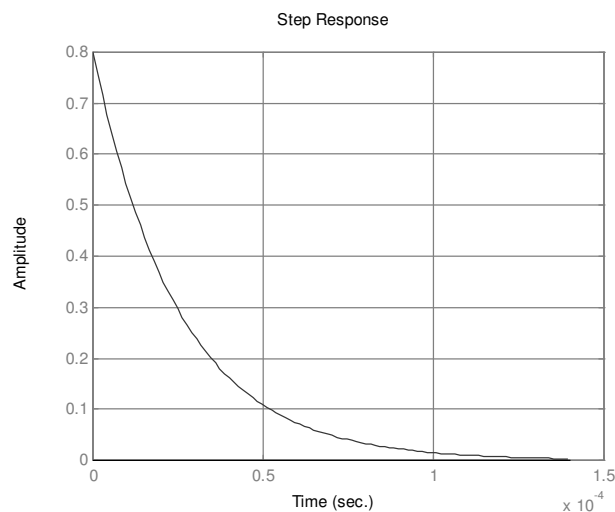
fasegradi =
 75.9638

Calcolo del valore di picco V_{oM} Volt della tensione di uscita

V_{oM} =
 0.9701

Risposta al gradino unitario

Premi un tasto per visualizzare il grafico



Esempio

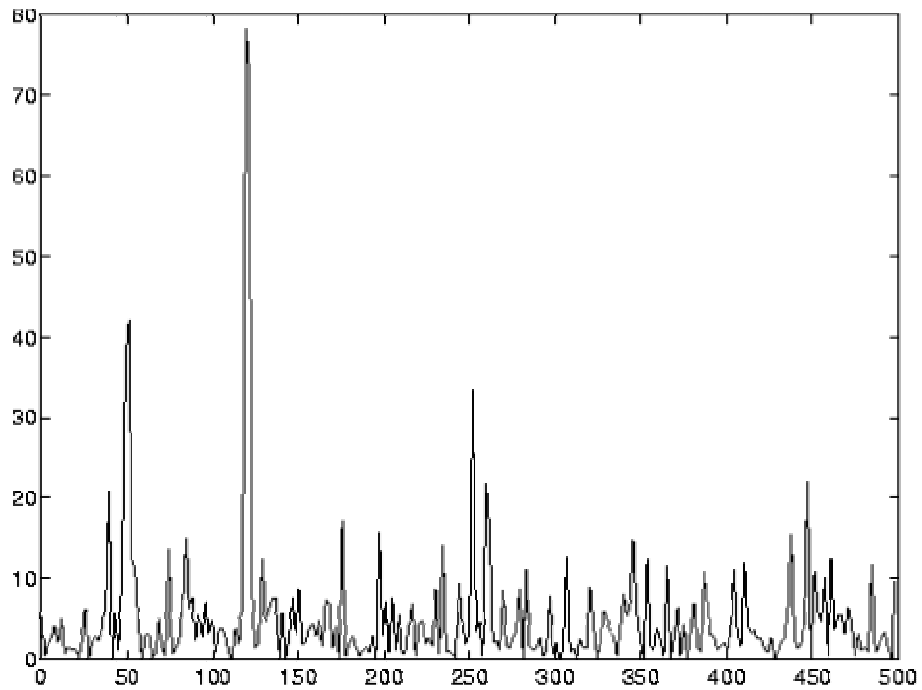
Si vuole determinare la potenza spettrale di un segnale

$y = x + 2 \cdot \text{randn}(1, \text{length}(t))$; costituito dalla somma di 2 segnali sinusoidali a 50 HZ e 120HZ e da un rumore a valor medio nullo.

Programma

```
t = 0:0.001:0.6;
x = sin(2*pi*50*t) + sin(2*pi*120*t);
y = x + 2*randn(1,length(t));
plot(y(1:50))
Y = fft(y,512); %FFT di y           %fast fourier trasform
Pyy = Y.*conj(Y) / 512;           % Densità di potenza spettrale dl segnale Y
f = 1000*(0:255)/512;
plot(f,Pyy(1:256))                %Grafico di 512 punti
```

Il software produce la seguente risposta



Esempio

Generare la funzione di trasferimento di un sistema del 2° ordine nota la pulsazione naturale e lo smorzamento. Sintassi $[num,den]= (wn,z)$. Nell'esempio $wn = 100$ e $z=0.1$

```
[num,den] = ord2(100,0.1)
sys = tf(num,den)
```

Si ottiene

```
num =
    1
den =
    1    2   100
sys = tf(num,den)
Transfer function:
    1
-----
s^2 + 2 s + 100
```

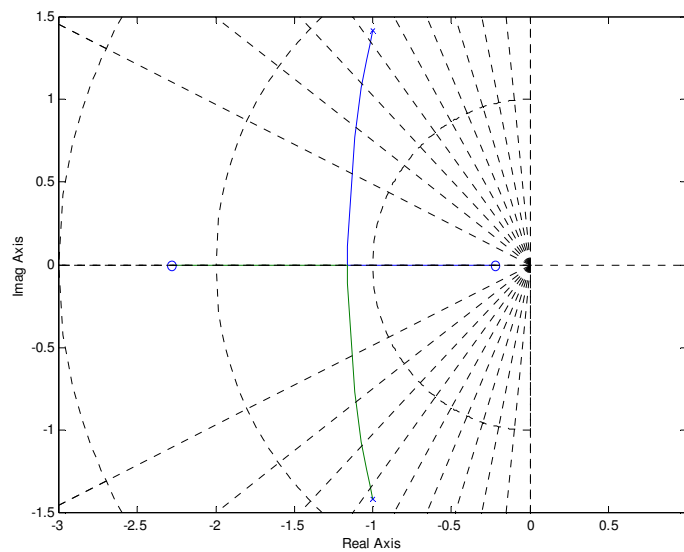
Esempio

% Luogo delle radici $G(s) = (2s^2 + 5s + 1)((s^2 + 2s + 3))$

```
H = tf([2 5 1],[1 2 3])
rlocus(H)
sgrid
```

Si ottiene

```
Transfer function:
    2 s^2 + 5 s + 1
-----
    s^2 + 2 s + 3
```



Esempio

Diagramma di Nichols

```
H = tf([-4 48 -18 250 600],[1 30 282 525 60])
```

```
nichols(H)
```

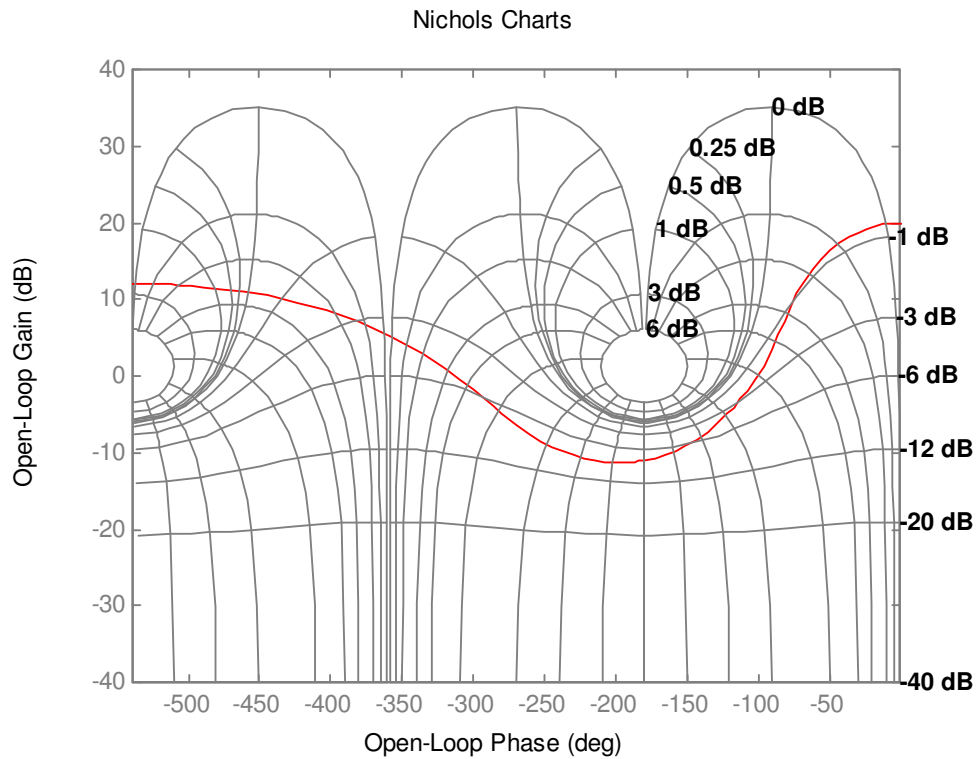
```
ngrid
```

Si ottiene

Transfer function:

$$-4 s^4 + 48 s^3 - 18 s^2 + 250 s + 600$$

$$s^4 + 30 s^3 + 282 s^2 + 525 s + 60$$

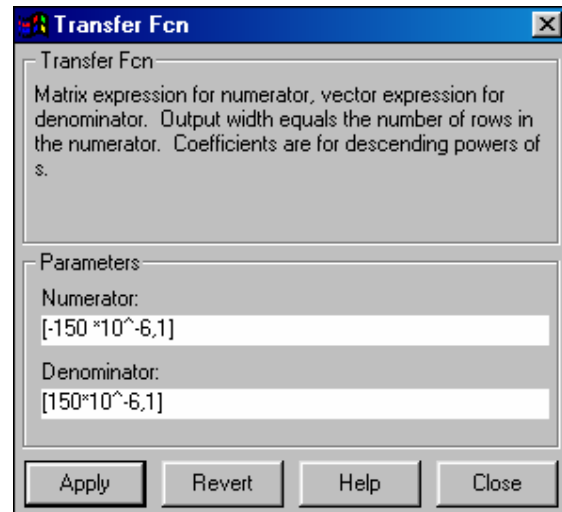
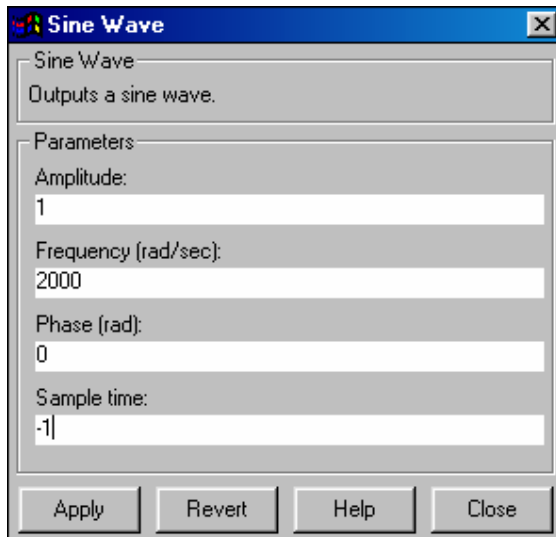
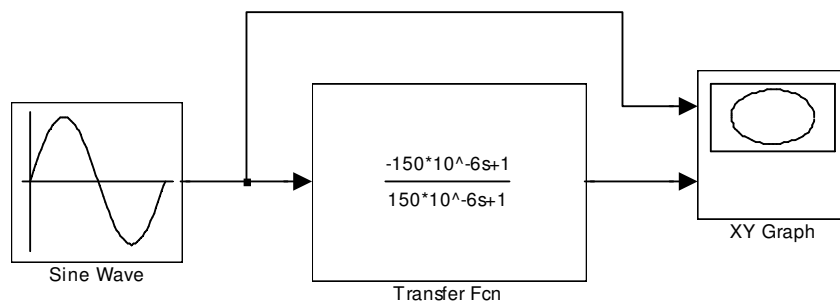


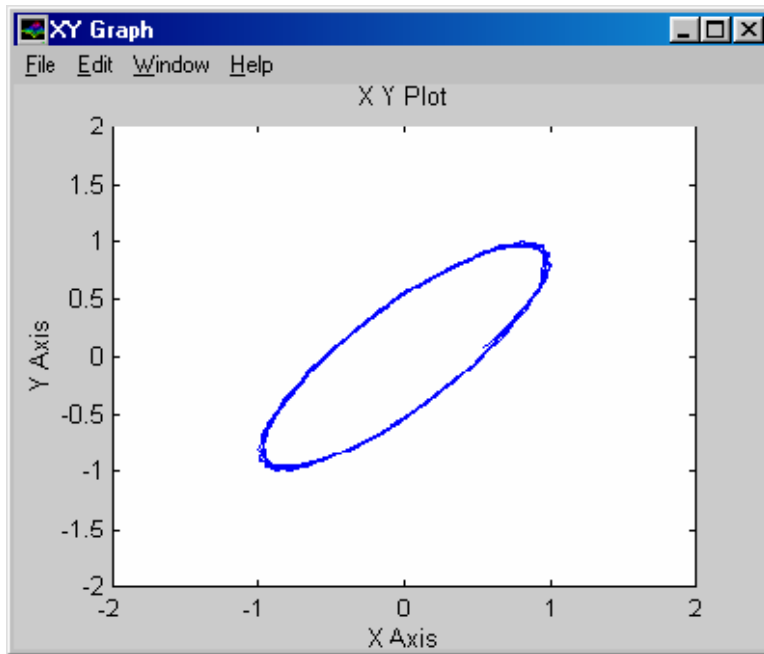
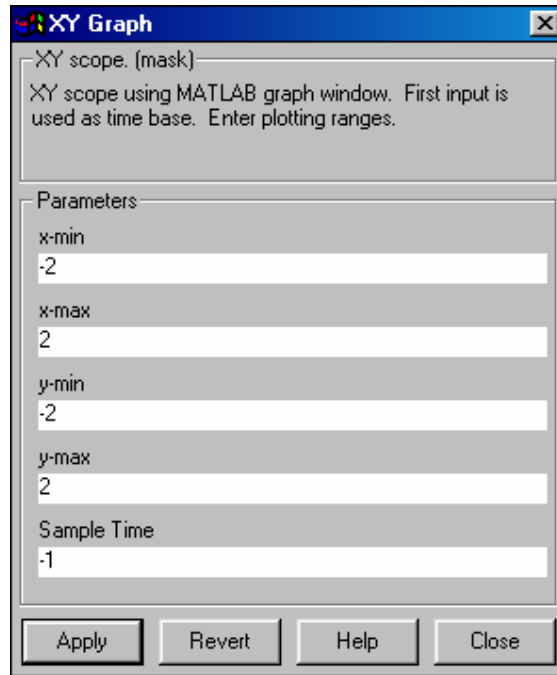
Esempio

Si riportano i blocchi di utilizzo in ambiente Simulink di Matlab per la simulazione del circuito sfasatore. La funzione di trasferimento vale:

$$G(s) = (1 - sRC)/(1 + sRC)$$

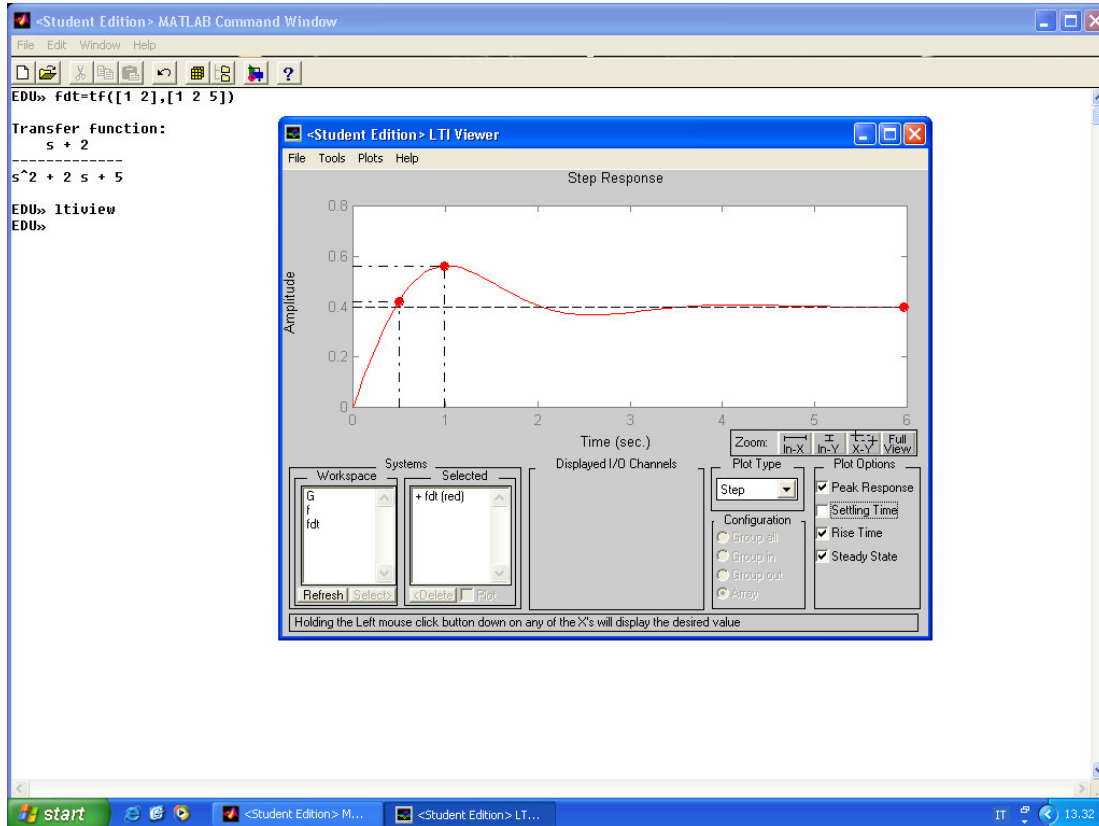
$$\text{con } RC = 150 \cdot 10^{-6}$$





Funzione Itview

Consente di esaminare la risposta nel dominio del tempo e nel dominio armonico di una f.d.t. Esempio



Quando si apre Itview cliccare 2 volte sulla funzione fdt nel Workspace. Scegliere il tipo di risposta nel Plot Type e le diverse opzioni du Plot Optino.