Analisi dei circuiti con MATLAB

1. INTRODUZIONE

MATLAB (MATrix LABoratory) è un programma interattivo per il calcolo numerico che offre numerosi strumenti utili per l'analisi dei circuiti. In questo primo paragrafo verranno brevemente richiamate alcune nozioni base di MATLAB per il calcolo e per la grafica.

Operazioni matematiche

Le operazioni matematiche fondamentali come addizione, sottrazione, moltiplicazione e divisione usano i simboli usuali `(+,-,*,/)`. L'elevamento a potenza utilizza l'operatore `^`. Questi simboli permettono di eseguire operazioni tra due scalari qualsiasi, reali o complessi. Si possono anche sommare, moltiplicare o dividere tutti gli elementi di un vettore per uno scalare. Ad esempio se `x` è un vettore, `x+1` aggiunge 1 a ciascun elemento di `x`; `x/2` divide ogni elemento di `x` per 2. Per elevare a potenza tutti gli elementi di un vettore occorre usare il simbolo `.^` (`x.^2` eleva al quadrato tutti gli elementi del vettore `x`).

Inoltre è possibile eseguire operazioni tra gli elementi di due vettori delle stesse dimensioni, come sommare gli elementi corrispondenti. Ad esempio se `x = [1 2 3]` e `y = [4 5 6]`, allora con il comando

```
w=x+y
```

si ottiene il vettore `w = [5 7 9]`.

Per moltiplicare due vettori, componente per componente, si fa precedere il simbolo `*` da un punto cioè `.*`. Ad esempio con i due vettori di prima si può scrivere

```
w=x.*y
```

ottenendo il vettore `w = [4 10 18]`.

Analogamente si può scrivere

```
w=y./x
```

dove `w = [4 2.5 2]`.

Con il comando

```
t=0:Δt:tmax;
```

si crea un vettore dei tempi, le cui componenti sono i valori ordinati da `0` a `tmax`, con intervallo pari a `Δt`. Ad esempio

```
t=0:0.01:10;
```

crea un vettore di 1001 componenti con intervallo 0.01, cioè $0, 0.01, 0.02, \cdots, 10$.

Dopo l'esecuzione di ogni comando viene automaticamente visualizzato il risultato (nell'esempio precedente le 1001 componenti del vettore). Per evitare ciò è necessario concludere il comando con un punto e virgola (;).

Le funzioni matematiche utili per calcolare la risposta dei circuiti lineari sono le seguenti:

esponenziale	`exp(...)`
seno	`sin(...)`
coseno	`cos(...)`
radice quadrata	`sqrt(...)`

Tali funzioni possono essere applicate ad un vettore; in tal caso restituiscono un vettore delle stesse dimensioni.

Ad esempio

```
t=0:0.01:10;
x=sin(t);
```

crea un vettore `x` di 1001 componenti.

Come ulteriore esempio, la corrente

$$i_R(t) = \frac{2}{\sqrt{3}} e^{-t/2} \cos\left(\frac{\sqrt{3}}{2}t + \frac{\pi}{6}\right)$$

per $0 \leq t \leq 10$ s può essere ottenuta con i seguenti comandi:

```
t=0:0.01:10;
IR=(2/sqrt(3))*exp(-t/2).*cos(t*sqrt(3)/2+pi/6);
```

I valori della corrente sono contenuti nel vettore IR di 1001 componenti.

Si noti come `exp(-t/2)` e `cos(t*sqrt(3)/2+pi/6)` siano due vettori delle stesse dimensioni di `t`. Per questo il loro prodotto si ottiene con l'operatore `.*`.

◆ Grafica

Per disegnare una funzione $f(t)$, occorre calcolare la funzione in una sequenza temporale di istanti $f(t_1), f(t_2), \ldots, f(t_{\max})$. Il grafico è ottenuto unendo questi valori con dei segmenti di retta (interpolazione lineare). Il comando per ottenere il grafico è `plot(x,y)`. Tale comando traccia segmenti di retta che uniscono i punti, le cui coordinate sono contenute nei vettori `x` e `y` (con dimensioni identiche).

Ad esempio

```
t=0:0.01:10;
y=sin(t);
plot(t,y)
```

traccia il grafico in Figura 1.1.

Si può modificare il tipo di linea includendo un terzo argomento nella funzione plot, ad esempio `plot(x,y,'--‘)` crea una linea tratteggiata.

Si possono tracciare diversi grafici utilizzando gli stessi assi. Per esempio

```
plot(x1,y1,x2,y2)
```

traccia `y1` in funzione di `x1` e `y2` in funzione di `x2` con gli stessi assi, utilizzando differenti tipi di linea.

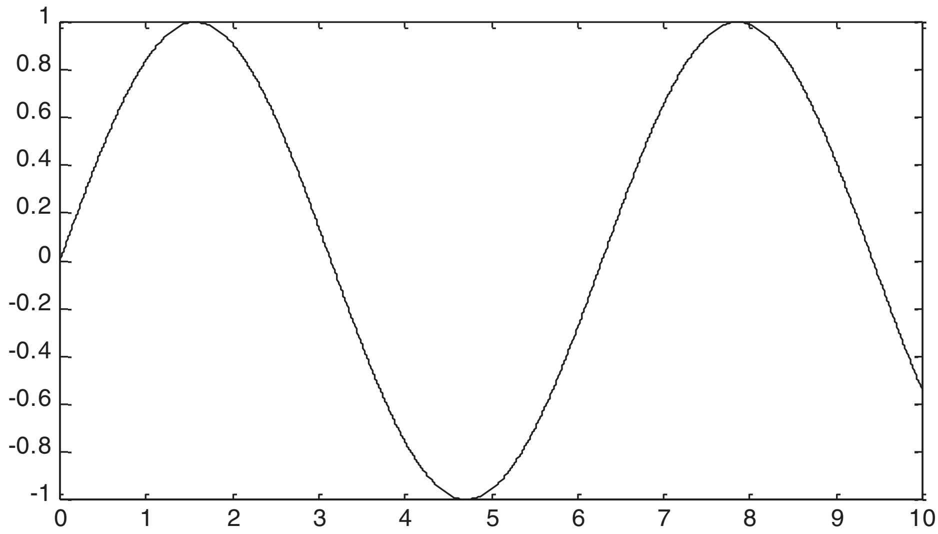


Figura 1.1

Aggiungendo `grid` al comando `plot` viene tracciata una griglia sul piano della figura (Fig. 1.2):

```
plot(t,y);grid
```

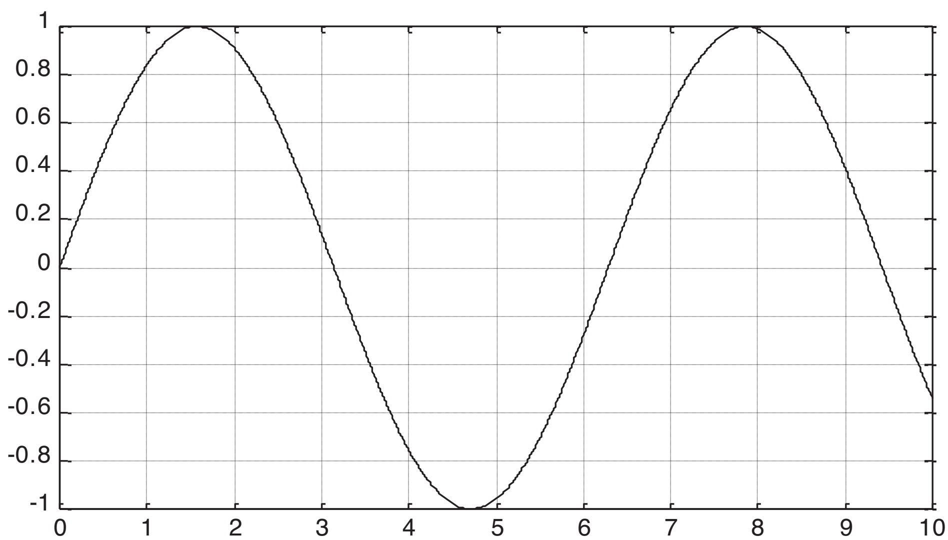


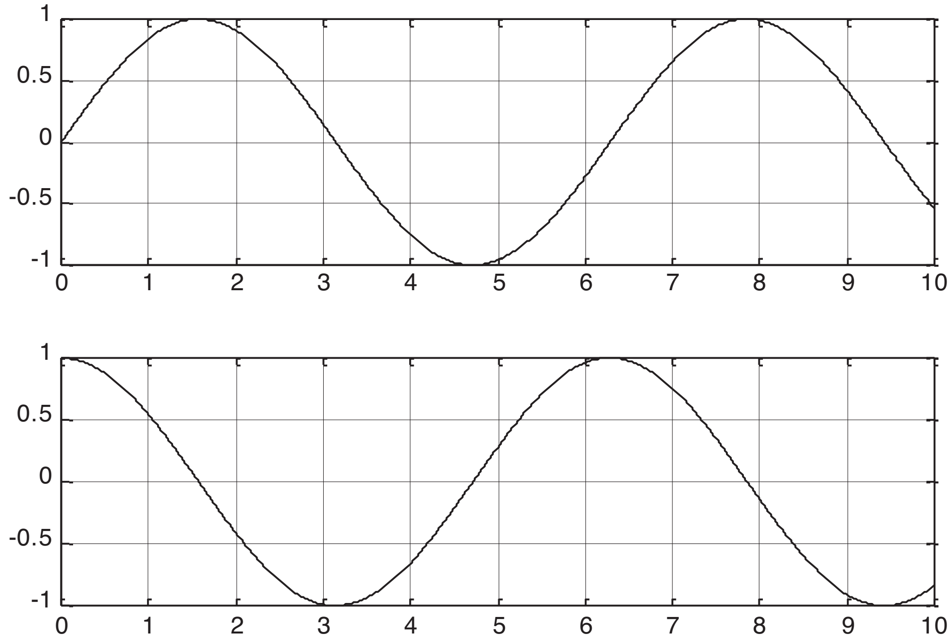
Figura 1.2

Un comando utile è `subplot` che permette di creare grafici distinti sulla stessa figura. Ad esempio con

```
t=0:0.01:10;
y=sin(t);
z=cos(t);
subplot(2,1,1);plot(t,y);grid
subplot(2,1,2);plot(t,z);grid
```

si crea la Fig. 1.3.

Figura 1.3



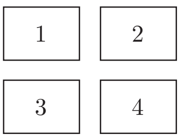


Figura 1.4

In generale `subplot(m,n,i)` crea l'i-esimo grafico di una matrice $m \times n$ di grafici. Al massimo si possono avere quattro grafici distinti. Nel caso $m = 2$ e $n = 2$ si hanno quattro grafici, disposti come in Figura 1.4.

La numerazione dei grafici è da sinistra verso destra e dall'alto verso il basso.

Per aggiungere un titolo alle figure si può usare

```
title('testo')
```

Per indicare il significato degli assi si usa

```
xlabel('testo')
ylabel('testo')
```

Per concludere vediamo il codice completo per disegnare le seguenti funzioni:

$$i_R(t) = \frac{2}{\sqrt{3}} e^{-t/2} \cos\left(\frac{\sqrt{3}}{2}t + \frac{\pi}{6}\right)$$

$$i_L(t) = 12 + e^{-t/2}\left[-\frac{1}{\sqrt{3}}\cos\left(\frac{\sqrt{3}}{2}t + \frac{\pi}{6}\right) + \operatorname{sen}\left(\frac{\sqrt{3}}{2}t + \frac{\pi}{6}\right)\right].$$

```
t=0:0.01:10;
q=(sqrt(3)/2)*t+pi/6;
IR=(2/sqrt(3))*exp(-t/2).*cos(q);
IL=12+exp(-t/2).*((-1/sqrt(3))*cos(q)+sin(q));
subplot(2,1,1);plot(t,IR);grid
xlabel('t,sec')
ylabel('iR(t), A')
subplot(2,1,2);plot(t,IL);grid
xlabel('t,sec')
ylabel('iL(t), A')
```

In Figura 1.5 sono mostrati i grafici corrispondenti.

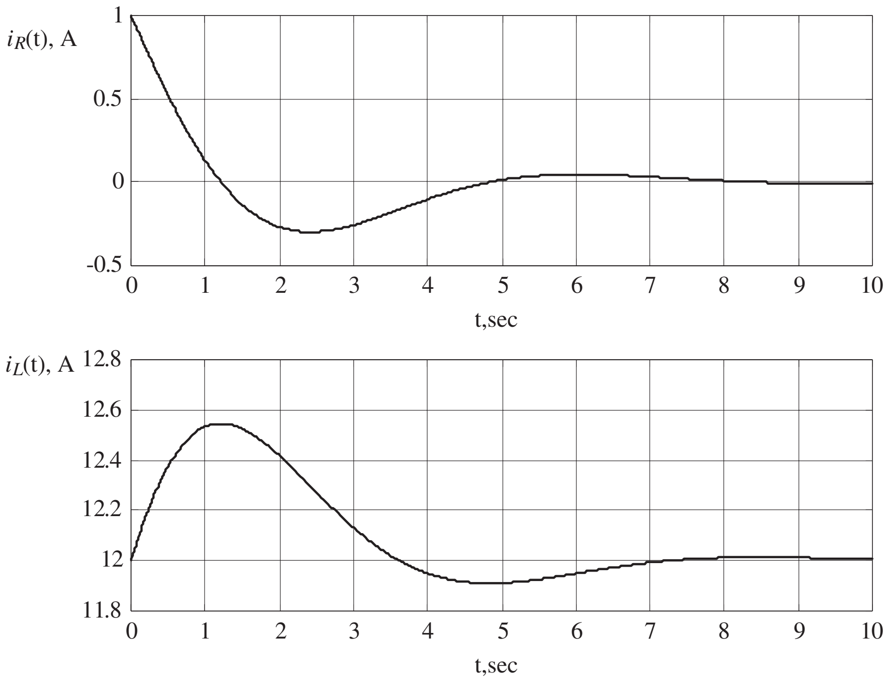


Figura 1.5

2. SOLUZIONE DI EQUAZIONI LINEARI

Utilizzando l'analisi nodale o delle maglie si perviene ad un sistema di equazioni lineari la cui complessità dipende dal numero di nodi o maglie del circuito. Nel caso di sistemi di ordine superiore al secondo per la soluzione conviene utilizzare strumenti software, come MATLAB. Il modo più semplice di inizializzare una matrice è attraverso il comando

```
nome_matrice=[a11 a12 a13 .....; a21 a22 a23.........;.......]
```

Per esempio

```
x = [1 2 3; 4 5 6]
```

crea la matrice

$$\mathbf{x} = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$$

Gli indici delle matrici seguono il seguente schema

$$\begin{matrix} (1,1) & (1,2) & (1,3) \\ (2,1) & (2,2) & (2,3) \\ (3,1) & (3,2) & (3,3) \end{matrix}$$

quindi il primo elemento ha indici (1,1) e non (0,0). Il primo elemento di un vettore ha sempre indice 1.

Un elemento di una matrice `x` può essere ottenuto scrivendo `x(i,j)`. Per esempio se `x` è la matrice 2×3 definita sopra allora

```
w = x(2,2)
```

fornisce `w = 5`.

I vettori possono essere costruiti come vettori riga o colonna. Per esempio

```
z = [1 2 3 4]
```

 © 978-88-08-17888-6

è un vettore riga, mentre

```
y = [1;2;3;4]
```

è il vettore colonna

$$\begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \end{bmatrix}$$

Con l'apice ′ si ottiene la trasposizione; nell'esempio precedente `y'` coincide con `z`.

La soluzione di un sistema di equazioni lineari $Ax = b$ si ottiene con l'operatore \ (*backslash o left matrix divide*) che corrisponde *formalmente* all'operazione $x = A^{-1}b$. Per esempio supponiamo di voler risolvere il sistema

$$\begin{bmatrix} 1 & 0 \\ 1 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 3 \end{bmatrix}$$

la cui soluzione è $x_1 = 1$, $x_2 = 1$.

I comandi MATLAB necessari sono i seguenti:

```
a = [1 0;1 2]
b = [1;3]
x = a\b
```

il risultato è

```
x =
1
1
```

Come ulteriore esempio vediamo la soluzione del sistema ottenuto nell'Esempio 4.1 del libro:

$$\begin{bmatrix} 0{,}7 & -0{,}5 & 0 \\ -0{,}5 & 1 & -0{,}5 \\ 0 & -0{,}5 & 0{,}75 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} = \begin{bmatrix} -2 \\ 3 \\ 2 \end{bmatrix}$$

MATLAB usa il punto decimale al posto della virgola. Nei risultati di MATLAB si userà quindi il punto decimale al posto della virgola

```
a=[0.7 -0.5 0;-0.5 1 -0.5;0 -0.5 0.75];
b=[-2;3;2];
v=a\b
```

con il risultato

```
v =
3.8462
9.3846
8.9231
```

Esercizi

2.1 Risolvere con MATLAB il sistema di equazioni seguente:

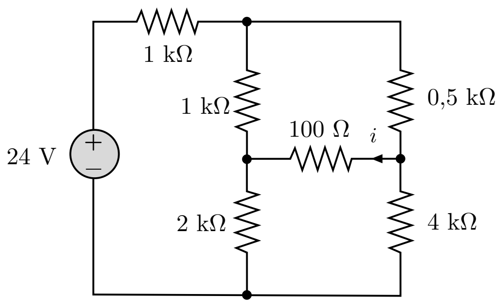
$$\begin{bmatrix} 2 & -0{,}5 & 1 \\ -0{,}5 & 1 & -1 \\ 3 & -0{,}5 & 2 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 2 \end{bmatrix}$$

$[v_1 = 0{,}4; v_2 = 0{,}8; v_3 = 0{,}6]$

2.2 Applicare l'analisi delle maglie al circuito in Figura 2.1. Scrivere il sistema di equazioni e risolverlo con MATLAB, quindi ricavare la corrente i.

$[i = 2{,}759\ \text{mA}]$

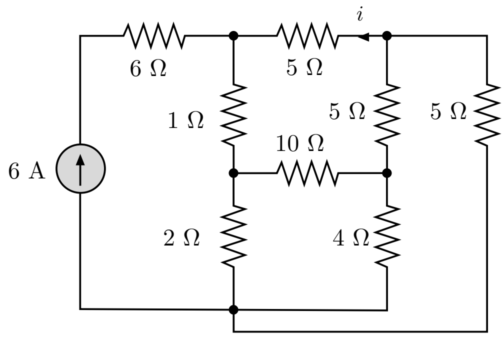
Figura 2.1



2.3 Per il circuito in Figura 2.2 scrivere il sistema di equazioni dell'analisi nodale per ispezione visiva. Risolvere il sistema con MATLAB, quindi ricavare la corrente i.

$[i = -1{,}461\ \text{A}]$

Figura 2.2



3. SOLUZIONE NUMERICA DELLE EQUAZIONI DI STATO

La soluzione numerica delle equazioni di stato si può ottenere facilmente con MATLAB utilizzando gli algoritmi di Eulero (capitolo 8). Il **metodo diretto di Eulero** per un circuito di ordine n corrisponde alla relazione (8.65) del libro:

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \Delta t(\mathbf{A}\mathbf{x}_k + \mathbf{u}_k) \qquad k = 0, 1, 2, \ldots \tag{3.1}$$

dove $\mathbf{x}$ è il vettore di stato con n componenti.

Assumendo un circuito del secondo ordine autonomo ($\mathbf{u}$ costante), le cui variabili di stato sono una tensione e una corrente, la (3.1) corrisponde ai seguenti comandi MATLAB:

```
..................
% metodo diretto di Eulero
for k=1:n
  x=x+delta*(A*x+u);
  v(k+1)=x(1);
  i(k+1)=x(2);
end
```

`x` è un vettore di appoggio di due componenti, mentre `v` ed `i` sono vettori di $n+1$ componenti, che contengono i valori delle due variabili di stato.

Si noti che nella relazione (3.1) k inizia da 0, mentre nel codice MATLAB il primo valore di k è uno; quindi `v(1)` e `i(1)` corrispondono alla soluzione in $t = 0$.

Il metodo inverso di Eulero corrisponde alla relazione

$$(\mathbf{I} - \Delta t\mathbf{A})\mathbf{x}_{k+1} = \mathbf{x}_k + \Delta t\mathbf{u}_{k+1} \tag{3.2}$$

dove con **I** si indica la matrice identità $n \times n$ (è la (8.66b) del libro).

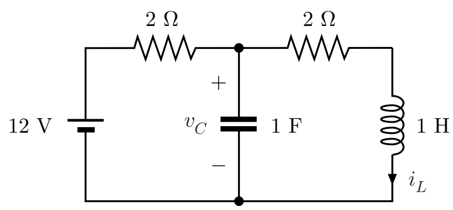
La (3.2) rappresenta un sistema di n equazioni lineari, le cui incognite sono le componenti del vettore $\mathbf{x}_{k+1}$. Il vettore dei termini noti è $(\mathbf{x}_k + \Delta t\mathbf{u}_{k+1})$. Risolvendo il sistema (3.2) per $k = 0, 1, 2, \ldots$, si ha la sequenza $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \ldots$

La soluzione si può ottenere con MATLAB utilizzando l'operatore *backslash* (\) (v. paragrafo 2). Supponendo **u** costante, per un circuito del secondo ordine si può utilizzare il seguente codice:

```
..................
% metodo inverso di Eulero
I=[1 0;0 1];
matrix=I-delta*A;
for k=1:n
b=x+delta*u;
  x=matrix\b;
  v(k+1)=x(1);
  i(k+1)=x(2);
end
```

Consideriamo il circuito dell'Esempio 8. 11 del libro (Fig. 3.1). Vogliamo calcolare la tensione $v_C(t)$ e la corrente $i_L(t)$. Le condizioni iniziali sono nulle.

Figura 3.1



Abbiamo

$$x = \begin{bmatrix} v_C \\ i_L \end{bmatrix} \quad \mathbf{A} = \begin{bmatrix} -0.5 & -1 \\ 1 & -2 \end{bmatrix} \quad u = \begin{bmatrix} 6 \\ 0 \end{bmatrix}$$

Il circuito è sottosmorzato; la costante di tempo con cui varia l'inviluppo della soluzione è $1/\alpha = 0.8$ s, perciò possiamo scegliere $\Delta t = 0.1$ s.

Di seguito si riporta il codice MATLAB utilizzato per ottenere la Figura 8.40 del libro.

```
% inizializzazione
n=100;
delta=0.1;
A=[-0.5 -1;1 -2];
u=[6;0];
% metodo diretto di Eulero
x=[0;0];
v(1)=x(1);
i(1)=x(2);
for k=1:n
  x=x+delta*(A*x+u);
  v(k+1)=x(1);
  i(k+1)=x(2);
end
% metodo inverso di Eulero
I=[1 0;0 1];
matrix=I-delta*A;
x=[0;0];
vs(1)=x(1);
is(1)=x(2);
for k=1:n
 b=x+delta*u;
  x=matrix\b;
  vs(k+1)=x(1);
  is(k+1)=x(2);
end
% costruzione vettore dei tempi
t=0:0.1:10;
% grafico della tensione
sol=6.41*exp(-1.25*t).*cos(0.66*t+160*pi/180)+6;
figure(1)
plot(t,v,'-.',t,vs,'--',t,sol,'-');grid
xlabel('tempo, secondi')
ylabel('v(t),volt')
% grafico della corrente
sol=6.42*exp(-1.25*t).*cos(0.66*t+117.83*pi/180)+3;
figure(2)
plot(t,i,'-.',t,is,'--',t,sol,'-');grid
xlabel('tempo, secondi')
ylabel('i(t),ampere')
```

4. ANALISI IN REGIME SINUSOIDALE

In questo paragrafo si descrive brevemente l'uso dei numeri complessi in MATLAB.

L'unità immaginaria può essere indicata indifferentemente con `i` oppure con `j`. Le seguenti espressioni sono equivalenti:

$$2+3\text{i} \quad 2+3\text{j} \quad 2+3{}^*\text{i} \quad 2+3{}^*\text{j}$$

I risultati dei calcoli riportano sempre la `i`.

Il modulo di un numero complesso `x` si ottiene con

```
abs(x)
```

l'argomento con

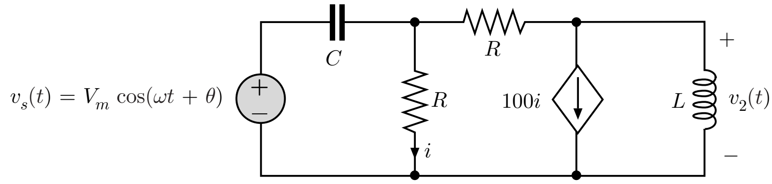
```
angle(x)
```

Il valore ottenuto con `angle` è espresso in radianti; per convertirlo in gradi si moltiplica per 180 e si divide per π(`pi`).

La soluzione dei sistemi di equazioni si ottiene come descritto nel *paragrafo 2*, anche quando i coefficienti sono complessi.

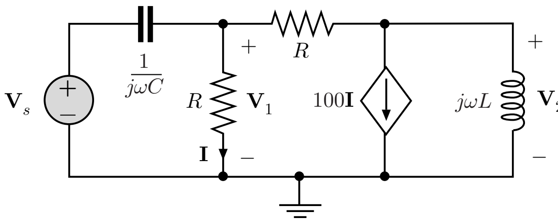
Come esempio, supponiamo di voler ricavare la tensione $v_2(t)$ in Figura 4.1.

Figura 4.1



Il circuito simbolico è riportato in Figura 4.2.

Figura 4.2



Con l'analisi nodale si ottiene il sistema seguente:

$$(\mathbf{V}_1-\mathbf{V}_s)j\omega C+\mathbf{V}_1\frac{1}{R}+(\mathbf{V}_1-\mathbf{V}_2)\frac{1}{R}=0$$
$$(\mathbf{V}_2-\mathbf{V}_1)\frac{1}{R}+100\mathbf{V}_1\frac{1}{R}+\mathbf{V}_2\frac{1}{j\omega L}=0$$

ovvero

$$\begin{bmatrix} j\omega C+\dfrac{2}{R} & -\dfrac{1}{R} \\ \dfrac{99}{R} & \dfrac{1}{R}+\dfrac{1}{j\omega L}\end{bmatrix}\begin{bmatrix}\mathbf{V}_1\\ \mathbf{V}_2\end{bmatrix}=\begin{bmatrix}j\omega C\mathbf{V}_s\\ 0\end{bmatrix} \tag{4.1}$$

Il codice MATLAB riportato di seguito risolve il sistema (4.1); si assumono i seguenti dati: $R = 1$ kΩ, $L = 10$ mH, $C = 2$ μF, $\omega = 1000$ rad/s, $V_m = 1$ V, $\theta = 0$.

```
%DATI
R=1000;
C=2*10^(-6);
L=10^(-2);
w=1000;
VS=1;

% AMMETTENZE
YC=j*w*C;
YL=1/(j*w*L);
% SISTEMA
A=[YC+2/R -1/R;99/R 1/R+YL];
B=[VS*YC;0];
% SOLUZIONE
X = A \ B
% modulo e fase di V2
A2=abs(X(2))
fi=angle(X(2))
fideg=fi*180/pi
```

I risultati sono:

```
X =
    0.4607 + 0.3097i
    0.3020 − 0.4591i
A2 =
    0.5496
fi =
    −0.9890
fideg =
    −56.6628
```

`A2` è l'ampiezza di $v_2(t)$, `fi` è la fase; `fideg` è la fase in gradi.

È possibile tracciare il grafico delle tensioni $v_s(t)$ e $v_2(t)$, aggiungendo al codice precedente i seguenti comandi:

```
%grafico di vs(t) e v2(t)
T=2*pi/w;
dt=T/1000;
Vm=abs(VS);
teta=angle(VS);
t=0:dt:2*T;
vs=Vm*cos(w*t+teta);
v2=A2*cos(w*t+fi);
plot(t,vs,'-',t,v2,'--');grid
xlabel('t,secondi')
title('vs- v2--')
```

T è il periodo delle due sinusoidi, pertanto il grafico viene tracciato per due periodi. L'intervallo di campionamento è stato scelto pari a $T/1000$. Il grafico è mostrato in Figura 4.3. La tensione del generatore indipendente è disegnata con linea continua; la tensione v_2 è tratteggiata. Si noti il periodo comune, pari a 6,3 ms, e il fatto che la tensione v_2 è in ritardo rispetto alla tensione v_s.

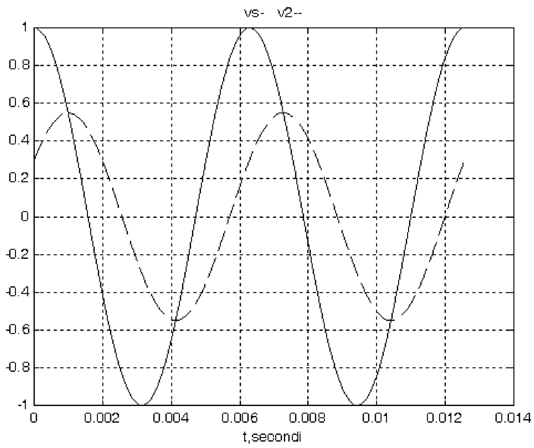


Figura 4.3

Esercizi

4.1 Risolvere l'Esercizio 9.40 del libro con MATLAB.

4.2 Risolvere l'Esercizio 9.60 del libro con MATLAB. Disegnare il grafico della soluzione.

5. RISPOSTA IN FREQUENZA

Per disegnare la risposta in frequenza conviene scrivere l'espressione della funzione di trasferimento nella forma razionale seguente:

$$\mathbf{H}(j\omega) = \frac{b_m(j\omega)^m + b_{m-1}(j\omega)^{m-1} + \ldots + b_1(j\omega) + b_0}{a_n(j\omega)^n + a_{n-1}(j\omega)^{n-1} + \ldots + a_1(j\omega) + a_0} \tag{5.1}$$

Si definiscono quindi i due vettori:

$$\mathtt{b} = [b_m b_{m-1} \ldots b_1 b_0] \qquad \mathtt{a} = [a_n a_{n-1} \ldots a_1 a_0]$$

Il comando

```
[H,W] = freqs (b,a)
```

restituisce due vettori di 200 componenti, `H` e `W`; `H` contiene i valori della funzione di trasferimento calcolati su una griglia di 200 pulsazioni equidistanti; i valori delle pulsazioni sono contenuti in `W`. Disegnando modulo e fase di `H` si ottiene la risposta in frequenza:

```
plot(W,abs(H))
plot(W,angle(H))
```

Il comando `angle` restituisce la fase in radianti; per averla in gradi:

```
plot(W,angle(H)*180/pi)
```

Infine, utilizzando

```
[H,W] = freqs(b,a,N)
```

si può scegliere il numero di punti (`N`) in cui viene calcolata la funzione di trasferimento.

Come esempio, consideriamo la funzione

$$\mathbf{H}(j\omega) = \frac{-10^4 j\omega}{(j\omega)^2 + 2\times 10^3 (j\omega) + 10^8} \tag{5.2}$$

in cui $m = 1$ ed $n = 2$. I vettori `b` ed `a` sono

`b` $= [-10^4 \quad 0]$
`a` $= [1 \quad 2\times 10^3 \quad 10^8]$.

Nel vettore `b` c'è una componente nulla; infatti i vettori `b` ed `a` devono avere sempre $m+1$ ed $n+1$ componenti, rispettivamente. Se scrivessimo `b` $= -10^4$ avremmo un risultato errato.

Per disegnare la risposta in frequenza utilizziamo il seguente codice:

```
b = [-10^4  0]
a = [1  2*10^3  10^8]
[H,W] = freqs(b,a);
subplot(2,1,1);plot(W,abs(H));grid
title('ampiezza')
subplot(2,1,2);plot(W,angle(H)*180/pi);grid
title('fase(gradi)')
```

I grafici sono mostrati in Figura 5.1.

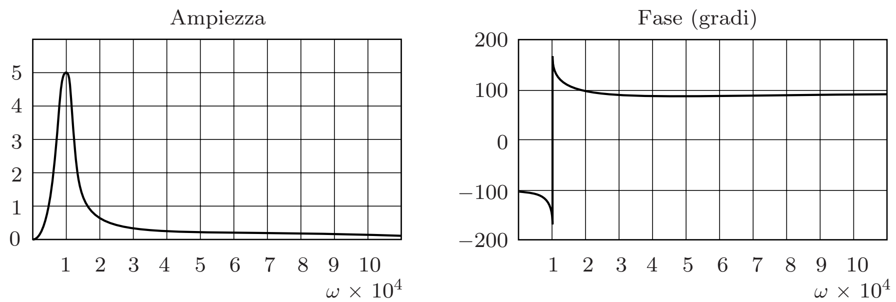


Figura 5.1 Risposte in ampiezza e in fase corrispondenti alla funzione (5.1).

Esercizi

5.1 Disegnare con MATLAB le risposte in ampiezza e in fase per il circuito dell'Esercizio 13.10 del libro, assumendo: $R_1 = 10$ kΩ, $R_2 = 5$ kΩ, $R = 1$ kΩ, $C = 0{,}02$ mF. La risposta è la tensione V.

5.2 Disegnare con MATLAB modulo e argomento della funzione di rete $\mathbf{Z}(j\omega)$ dell'Esercizio 13.45 del libro.

5.3 Disegnare con MATLAB le risposte in ampiezza e in fase per la funzione di rete seguente e confrontarli con i diagrammi di Bode ottenuti manualmente:

$$\mathbf{F}(j\omega) = \frac{100(j\omega+2)}{(j\omega+1)(j\omega+10)^2}$$

5.4 Disegnare con MATLAB le risposte in ampiezza e in fase per la funzione di rete seguente e confrontarli con i diagrammi di Bode ottenuti manualmente:

$$\mathbf{F}(j\omega) = \frac{j\omega 100}{(j\omega)^2 + (j\omega)100 + 16\times 10^4}\,.$$

6. TRASFORMATA DI LAPLACE

Lo sviluppo in frazioni parziali di una funzione razionale reale può essere agevolmente ottenuto con MATLAB. La funzione da antitrasformare sia:

$$\mathbf{F}(s) = \frac{\mathbf{b}(s)}{\mathbf{a}(s)} = \frac{b_m s^m + b_{m-1}s^{m-1} + \ldots + b_1 s + b_0}{a_n s^n + a_{n-1}s^{n-1} + \ldots + a_1 s + a_0} \tag{6.1}$$

Nel caso di **poli semplici** lo sviluppo in frazioni parziali è

$$\mathbf{F}(s) = \mathbf{q}(s) + \frac{A_1}{s - p_1} + \frac{A_2}{s - p_2} + \ldots + \frac{A_n}{s - p_n} \tag{6.2}$$

dove $\mathbf{q}(s)$ è un polinomio di grado $m - n$.

I poli p_k, i residui A_k e i coefficienti del polinomio $\mathbf{q}(s)$ si possono ottenere con il comando `residue`. A tale scopo definiamo i due vettori:

$$\mathtt{b} = [b_m b_{m-1} \ldots b_1 b_0] \qquad \mathtt{a} = [a_n a_{n-1} \ldots a_1 a_0]$$

Il comando

```
[r,p,k] = residue(b,a)
```

fornisce tre vettori: `r` contiene i valori dei *residui*, `p` i valori dei *poli* corrispondenti, e infine `k` i *coefficienti del polinomio* $\mathbf{q}(s)$.
Il comando

```
[b,a] = residue(r,p,k)
```

effettua l'operazione inversa, fornendo i coefficienti del numeratore e del denominatore, a partire dai poli e dai residui. MATLAB distingue i due comandi in base al numero di argomenti di ingresso e di uscita.

Per esempio, per ottenere lo sviluppo in frazioni parziali della funzione:

$$\mathbf{F}(s) = \frac{2s^2 + s + 3}{6s^2 + s + 9}$$

si utilizzano i seguenti comandi:

```
b = [2  1  3];
a = [6  1  9];
[r,p,k] = residue(b,a)
```

Il risultato è

```
r =
     0.0556 + 0.0038i
     0.0556 - 0.0038i
p =
    -0.0833 + 1.2219i
    -0.0833 - 1.2219i
k =
     0.3333
```

pertanto lo sviluppo è,

$$\mathbf{F}(s) = 0.3333 + \frac{0.0556 + j0.0038}{s + 0.0833 - j1.2219} + \frac{0.0556 - j0.0038}{s + 0.0833 + j1.2219}$$

Utilizzando

```
[b,a] = residue(r,p,k)
```

si ottiene

```
b =
    0.3333  0.1667  0.5000

a =
    1.0000  0.1667  1.5000
```

ovvero i coefficienti dei due polinomi, **normalizzati** rispetto ad $a_3 = 6$.

I coefficienti di $\mathbf{b}(s)$ e $\mathbf{a}(s)$ che corrispondono ad eventuali potenze mancanti vanno posti a zero. Per esempio lo sviluppo della funzione

$$\mathbf{F}(s) = \frac{2s^2 + 3}{s^2 + 4s + 3}$$

si ottiene con

```
b = [2  0  3];
a = [1  4  3];
[r,p,k] = residue(b,a)
```

il risultato è

```
r =
    -10.5000
      2.5000

p =
    -3
    -1

k =
    2
```

Lo sviluppo in frazioni parziali è

$$\mathbf{F}(s) = 2 + \frac{-10.5}{s+3} + \frac{2.5}{s+1}$$

Il comando `residue` può essere usato anche nel caso di *poli multipli*. Se invece vogliamo conoscere solamente i poli della funzione $\mathbf{F}(s)$ possiamo usare il comando:

```
p = roots(a)
```

Esercizi

6.1 Verificare le soluzioni dell'Esercizio 14.18 del libro con MATLAB.

6.2 Verificare le soluzioni dell'Esercizio 14.19 del libro con MATLAB.