

Sistemi lineari in Matlab

● Esempio numerico in Matlab: LTI_Systems_in_Matlab.m

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   LTI_Systems_in_Matlab.m
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clear all; clc
echo on
% Si consideri il sistema lineare (LS) tempo continuo (TC) caratterizzato
% dalle seguenti matrici MA, MB e MC
% MA=[ 2    3    3;
%     -4   -5   -3;
%     -2    1   -1];
% MB=[ 3; -2; 1];
% MC=[ 2  1  0];
echo off
MA=[ 2    3    3;
     -4   -5   -3;
     -2    1   -1];
MB=[ 3; -2; 1];
MC=[ 2  1  0];
echo on
pause; clc;           % Premi un tasto per continuare
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Per definire un sistema tempo continuo in ambiente Matlab si utilizza il
% comando "ss" (state space). Utilizzando l'istruzione
SYS=ss(MA,MB,MC,0)

----- Matlab output -----
SYS =

a =
    x1  x2  x3
x1    2    3    3
x2   -4   -5   -3
x3   -2    1   -1

b =
    u1
x1    3
x2   -2
x3    1

c =
    x1  x2  x3
y1    2    1    0

d =
    u1
y1    0
-----

% Si assegna alla variabile SYS il significato di sistema dinamico lineare
% caratterizzato dalle 3 matrici MA, MB e MC (la matrice MD=0).
pause; clc;           % Premi un tasto per continuare
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% E' possibile assegnare un nome alle variabili di ingresso, di stato e di uscita
% utilizzando il comando "set"
set(SYS,'InputName','Tensione')
set(SYS,'StateName',['Corrente '; 'Velocita '; 'Posizione'])
set(SYS,'OutputName','Uscita')
SYS

```

```

----- Matlab output -----
SYS =

a =
      Corrente  Velocita  Posizione
Corrente      2         3         3
Velocita     -4        -5        -3
Posizione    -2         1        -1

b =
      Tensione
Corrente      3
Velocita     -2
Posizione     1

c =
      Corrente  Velocita  Posizione
Uscita       2         1         0

d =
      Tensione
Uscita       0
-----

pause; clc;          % Premi un tasto per continuare
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% La funzione di trasferimento del sistema G(s) si ottiene utilizzando il comando "tf"
GS=tf(SYS)
----- Matlab output -----
GS =

From input "Tensione" to output "Uscita":
      4 s^2 + 17 s + 28
-----
      s^3 + 4 s^2 + 14 s + 20
-----

pause; clc;          % Premi un tasto per continuare
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% I poli del sistema $G(s)$
pole(GS)
----- Matlab output -----
ans =

-1.0000 + 3.0000i
-1.0000 - 3.0000i
-2.0000
-----

% coincidono con i poli del sistema SYS
pole(SYS)
----- Matlab output -----
ans =

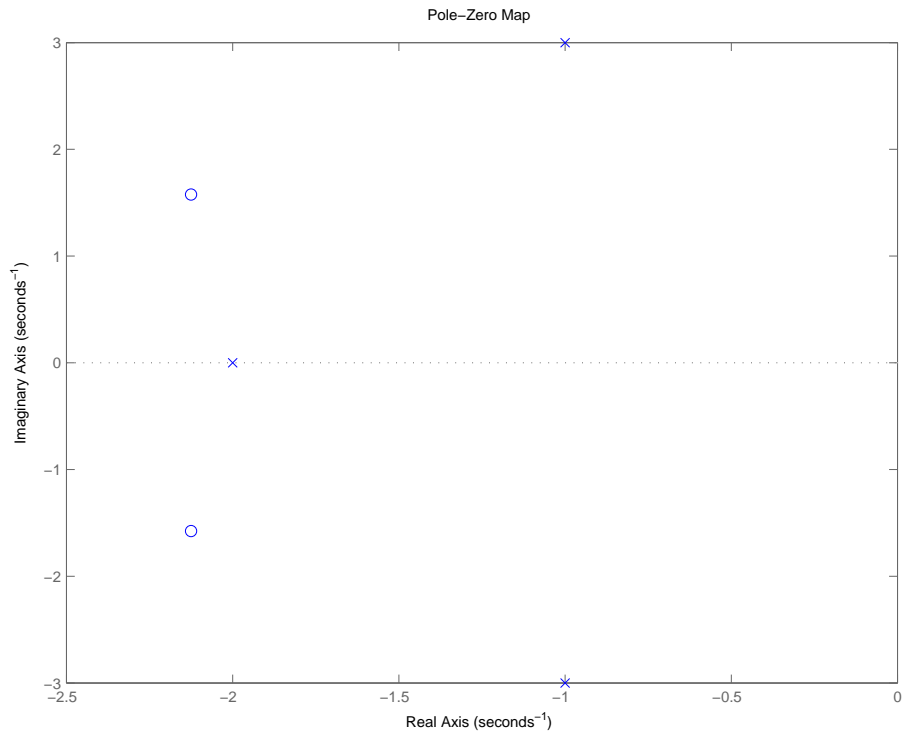
-1.0000 + 3.0000i
-1.0000 - 3.0000i
-2.0000
-----

% e coincidono con gli autovalori della matrice MA
roots(poly(MA))
----- Matlab output -----
ans =

-1.0000 + 3.0000i
-1.0000 - 3.0000i
-2.0000
-----

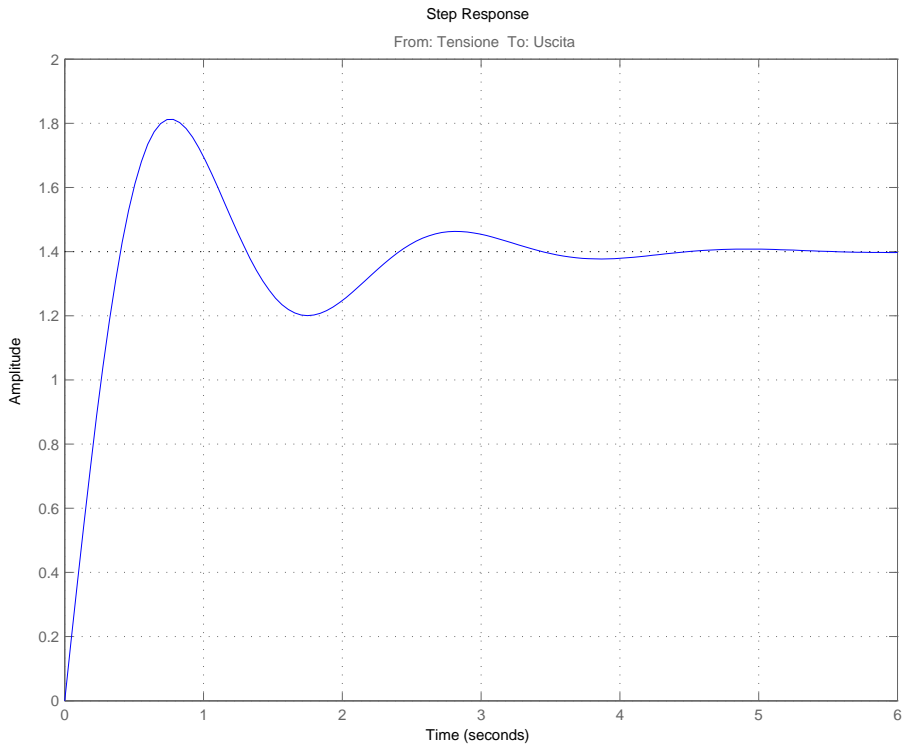
pause; clc;          % Premi un tasto per continuare
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% La posizione dei poli e degli zeri del sistema SYS possono essere
% visualizzati utilizzando l'istruzione
figure(1); clf
pzmap(SYS)
----- Matlab output -----

```



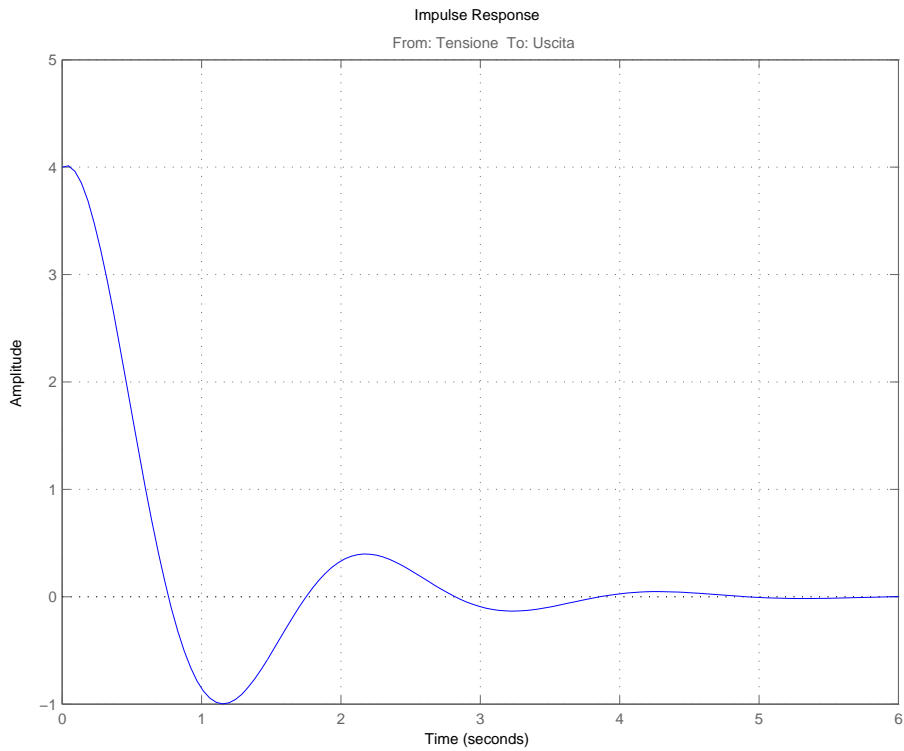
```

-----
pause; % Premi un tasto per continuare
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% La risposta al gradino si ottiene utilizzando il comando
figure(1); clf
step(SYS); grid on
----- Matlab output -----
    
```



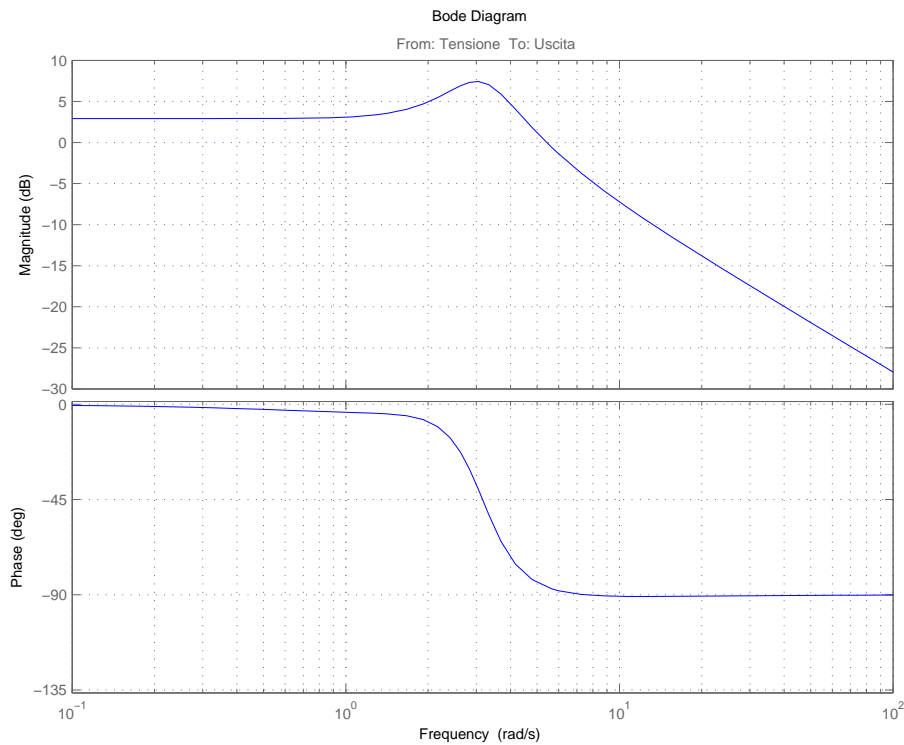
```

-----
pause; % Premi un tasto per continuare
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% La risposta all'impulso si ottiene utilizzando il comando
figure(1); clf
impulse(SYS); grid on
----- Matlab output -----
    
```



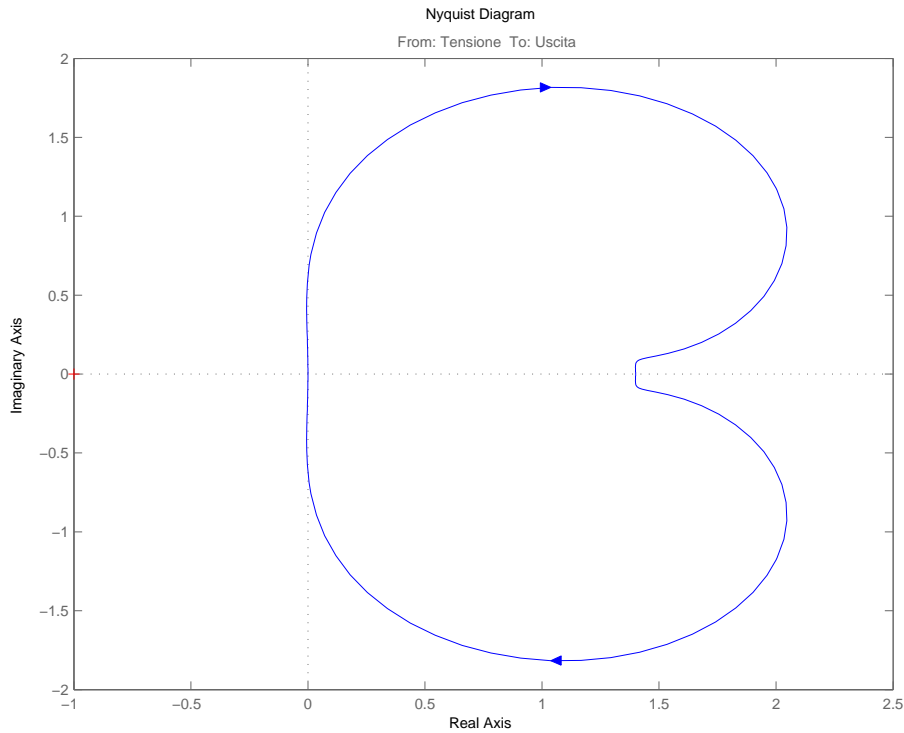
```

-----
pause; % Premi un tasto per continuare
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% I diagrammi di Bode si ottengono utilizzando il comando:
figure(1); clf
bode(SYS); grid on
----- Matlab output -----
    
```



```

-----
pause; % Premi un tasto per continuare
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% I diagrammi di Nyquist si ottengono utilizzando il comando:
figure(1); clf
nyquist(SYS)
----- Matlab output -----
    
```



```

-----
pause;                % Premi un tasto per continuare
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
set(gcf,'Visible','off')
clc
% Gli autovalori e gli autovettori della matrice MA si ottengono
% utilizzando il seguente comando
[V,D] = eig(MA)
% I primi due autovettori (cioe' le prime due colonne della matrice V)
% sono complessi coniugati perche' sono gli autovettori corrispondenti
% ai due autovalori complessi coniugati. La matrice V la matrice di
% trasformazione che permette di diagonalizzare la matrice $MA$
inv(V)*MA*V
% Trascurando i piccoli errori di calcolo si ottiene:
approx(inv(V)*MA*V)

```

----- Matlab output -----

```

% Gli autovalori e gli autovettori della matrice MA si ottengono
% utilizzando il seguente comando
[V,D] = eig(MA)

```

```

V =

    0.5774         0.5774         -0.0000
   -0.5774 - 0.0000i   -0.5774 + 0.0000i   -0.7071
   -0.0000 + 0.5774i   -0.0000 - 0.5774i    0.7071

```

```

D =

   -1.0000 + 3.0000i         0         0
         0         -1.0000 - 3.0000i         0
         0         0         -2.0000

```

```

% I primi due autovettori (cioe' le prime due colonne della matrice V)
% sono complessi coniugati perche' sono gli autovettori corrispondenti
% ai due autovalori complessi coniugati. La matrice V la matrice di
% trasformazione che permette di diagonalizzare la matrice $MA$
inv(V)*MA*V

```

```

ans =

   -1.0000 + 3.0000i   -0.0000 + 0.0000i   -0.0000 - 0.0000i
   -0.0000 - 0.0000i   -1.0000 - 3.0000i   -0.0000 + 0.0000i
    0.0000 + 0.0000i    0.0000 - 0.0000i   -2.0000 + 0.0000i

```

```
% Trascurando i piccoli errori di calcolo si ottiene:
approx(inv(V)*MA*V)

ans =

    -1.0000 + 3.0000i     0     0
         0    -1.0000 - 3.0000i     0
         0         0    -2.0000

-----

pause; clc;           % Premi un tasto per continuare
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Per portare la matrice MA in forma reale di Jordan, occorre utilizzare
% una matrice di trasformazione reale: gli autovettori complessi coniugati
% vengono sostituiti dai vettori parte-reale e parte-immaginaria di uno
% dei due autovettori complessi coniugati
TM=[real(V(:,1)) imag(V(:,1)) V(:,3)]
inv(TM)*MA*TM
% Trascurando i piccoli errori di calcolo si ottiene:
approx(inv(TM)*MA*TM)

----- Matlab output -----
% Per portare la matrice MA in forma reale di Jordan, occorre utilizzare
% una matrice di trasformazione reale: gli autovettori complessi coniugati
% vengono sostituiti dai vettori parte-reale e parte-immaginaria di uno
% dei due autovettori complessi coniugati
TM=[real(V(:,1)) imag(V(:,1)) V(:,3)]

TM =

    0.5774     0    -0.0000
   -0.5774   -0.0000   -0.7071
   -0.0000    0.5774    0.7071

inv(TM)*MA*TM

ans =

   -1.0000    3.0000   -0.0000
   -3.0000   -1.0000    0.0000
    0.0000    0.0000   -2.0000

% Trascurando i piccoli errori di calcolo si ottiene:
approx(inv(TM)*MA*TM)

ans =

    -1     3     0
    -3    -1     0
     0     0    -2

-----

pause; clc;           % Premi un tasto per continuare
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Alla stessa forma canonica reale di Jordan si poteva giungere
% utilizzando il comando
[CSYS, TM]=canon(SYS)

----- Matlab output -----
% Alla stessa forma canonica reale di Jordan si poteva giungere
% utilizzando il comando
[CSYS, TM]=canon(SYS)

CSYS =

a =

      Corrente  Velocita  Posizione
Corrente      -1         3         0
Velocita      -3        -1         0
Posizione      0         0        -2

b =

      Tensione
Corrente       2
Velocita      -3
Posizione     -1.414
```

```
c =
    Corrente  Velocita  Posizione
Uscita      0        -1   -0.7071

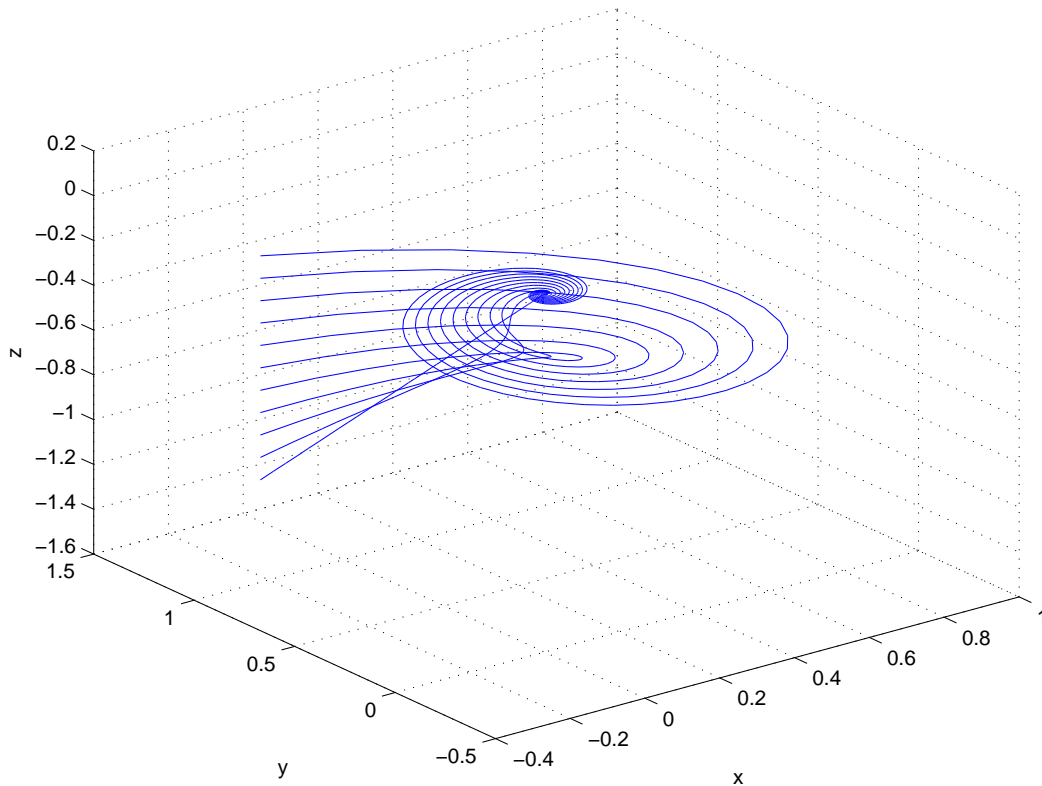
d =
    Tensione
Uscita      0
```

Continuous-time state-space model.

```
TM =
    1.0000    1.0000    1.0000
   -1.0000    0.0000    0.0000
   -1.4142   -1.4142   -0.0000
```

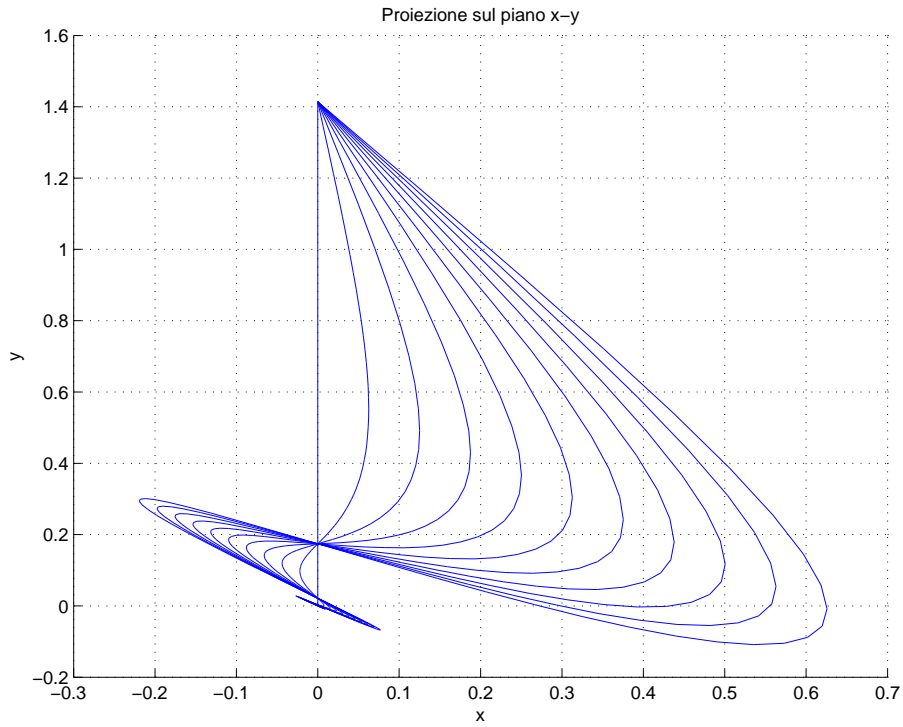
```
-----
pause; clc; % Premi un tasto per continuare
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% L'evoluzione libera del sistema si ottiene con il comando "initial":
% X0 stato iniziale; X e Y vettori di stato e di uscita ; T vettore dei tempi
figure(1); clf
X00=-2*V(:,3);
dx0=[0:0.1:1];
for dx=dx0
    X0=X00+[0;0;dx];
    [Y,T,X] = initial(SYS,X0);
    plot3(X(:,1),X(:,2),X(:,3)); hold on
    echo off
end
grid; xlabel('x'); ylabel('y'); zlabel('z')
echo on
```

----- Matlab output -----



```
-----
pause; % Premi un tasto per continuare
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Proiezione sul piano x-y
figure(1)
view(0,90)
title('Proiezione sul piano x-y')
```

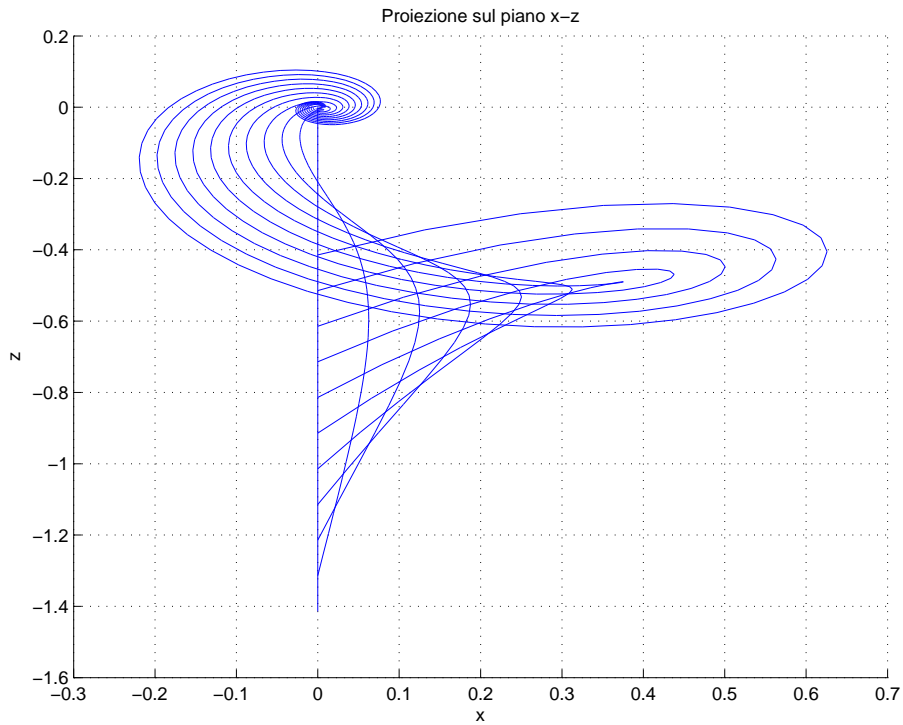
----- Matlab output -----



```

pause; % Premi un tasto per continuare
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Proiezione sul piano x-z
figure(1)
view(0,0)
title('Proiezione sul piano x-z')
    
```

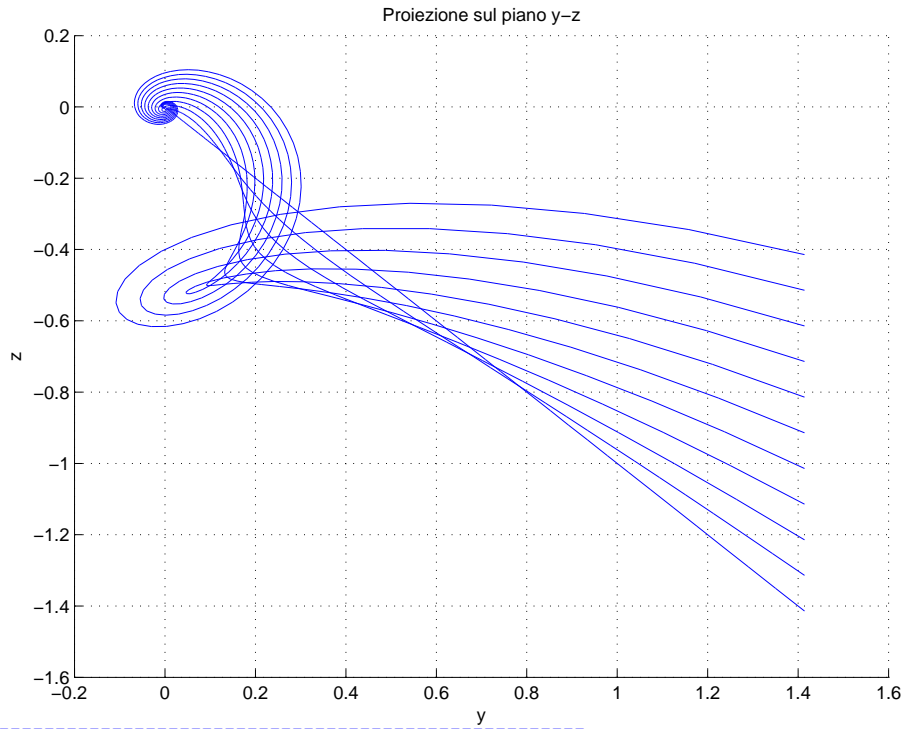
----- Matlab output -----



```

pause; % Premi un tasto per continuare
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Proiezione sul piano y-z
figure(1)
view(90,0)
title('Proiezione sul piano y-z')
    
```

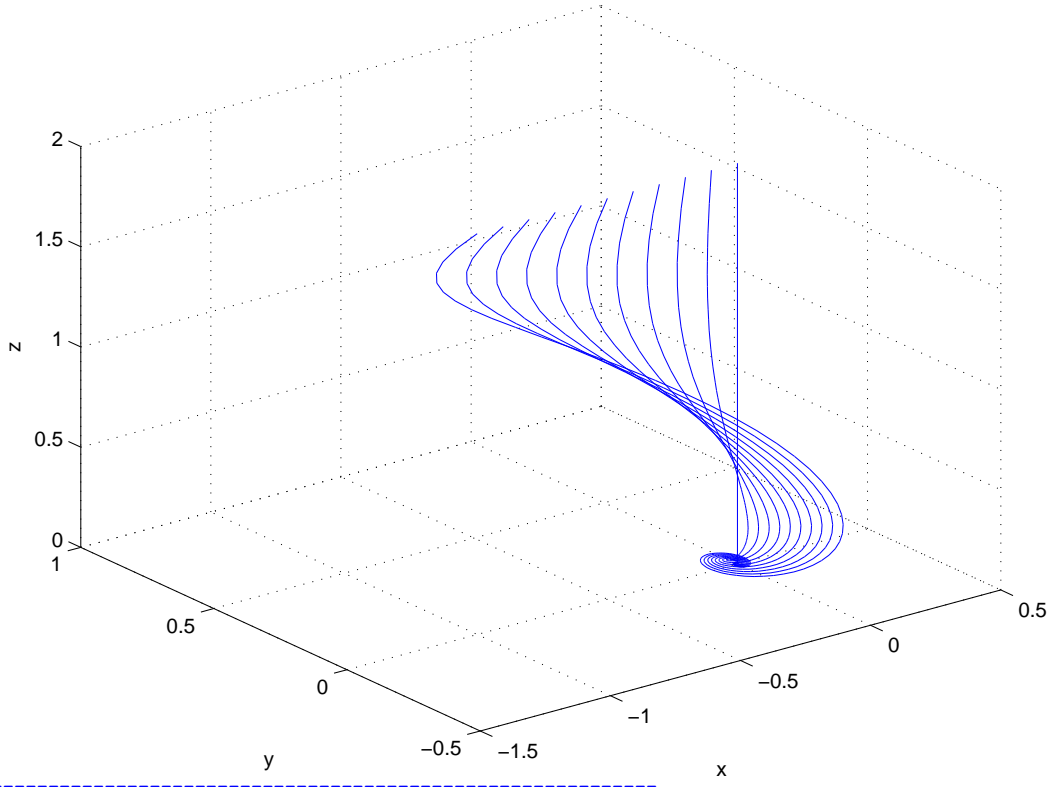
----- Matlab output -----



```

pause; clc; % Premi un tasto per continuare
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Le stesse traiettorie nella base canonica reale di Jordan
% assumono la seguente forma:
figure(1); clf
for dx=dx0
    X0=X00+[0;0;dx];
    [Y,T,X] = initial(CSYS,-TM*X0);
    plot3(X(:,1),X(:,2),X(:,3)); hold on
    echo off
end
grid; xlabel('x'); ylabel('y'); zlabel('z')
echo on
    
```

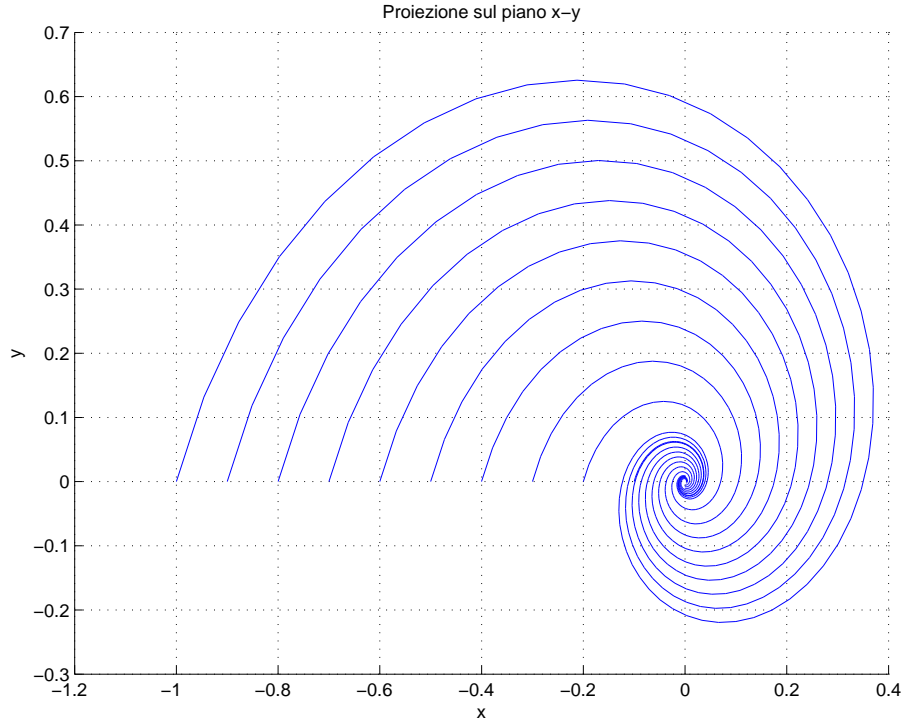
----- Matlab output -----



```

pause; % Premi un tasto per continuare
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Proiezione sul piano x-y
figure(1)
view(0,90)
title('Proiezione sul piano x-y')
    
```

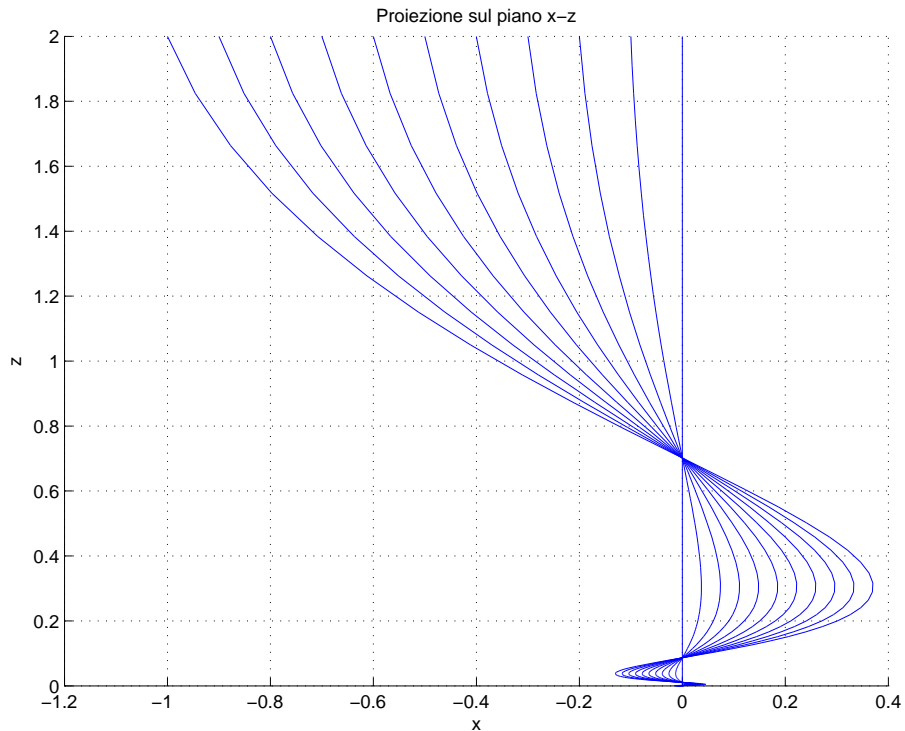
----- Matlab output -----



```

pause; % Premi un tasto per continuare
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Proiezione sul piano x-z
figure(1)
view(0,0)
title('Proiezione sul piano x-z')
    
```

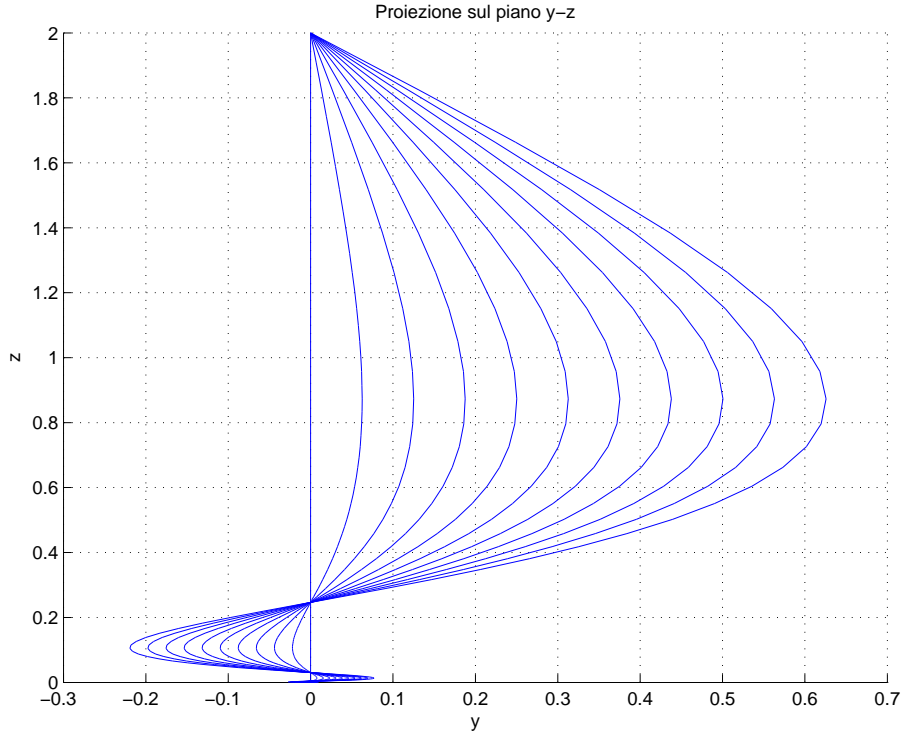
----- Matlab output -----



```

pause; % Premi un tasto per continuare
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Proiezione sul piano y-z
figure(1)
view(90,0)
title('Proiezione sul piano y-z')
    
```

----- Matlab output -----



echo off