MATLAB

- Dedicato all'analisi, la simulazione e la sintesi di sistemi lineari
- Orientato all'impiego di tecniche di algebra lineare
- Costituito da un "motore" e diversi toolbox per affrontare problematiche specifiche (controllistiche, di signal processing, di stima, ecc.)

Nella versione più completa è
- uno shell (risponde a comandi immediati)
- un interprete di script
- un interprete/simulatore di schemi a blocchi

(in attesa di scrivere quelle per SCILAB, si impieghino queste, vista la similarità d'uso dei due programmi)

Variabili e operatori

-Una variabile può essere uno scalare o una matrice
-Per default, tutte le grandezze sono complesse
-Gli operatori elementari si applicano anche alle matrici
 (es.: 2*3 e A*B)
- Esiste la forma “elemento per elemento” col punto prefisso
  (es.: A*B e A.*B)

- Sono presenti molte funzioni dedicate
  (es.: step(num, den, t) traccia la risposta a gradino)

Radici

```
» poly = [5 4 3 -2 -1];
» roots(poly)

ans =

  -0.5000 + 0.8660i
  -0.5000 - 0.8660i
   0.5583
  -0.3583
```

Una FdT

$$G(s) = \frac{s+2}{(s+3)(2s+10)}$$

```
» num=[1 2]
num =
     1     2

» den=conv([1 3],[2 10])
den =
     2    16    30
```

Frazioni parziali

```
» [r,p,li]=residue(num,den)
r =
    0.7500
   -0.2500
p =
    -5
    -3
li =
     [ ]
```

$$G(s) = \frac{0.75}{s+5} + \frac{-0.25}{s+3}$$

```
>> n1=1; d1=[1 1];
>> n2=[3 1]; d2=[10 1];
```

Due sistemi

```
>> [n,d]=series(n1,d1,n2,d2)
n =
   0   3   1
d =
  10  11   1
```

```
>> [n,d]=parallel(n1,d1,n2,d2)
n =
   3  14   2
d =
  10  11   1
```

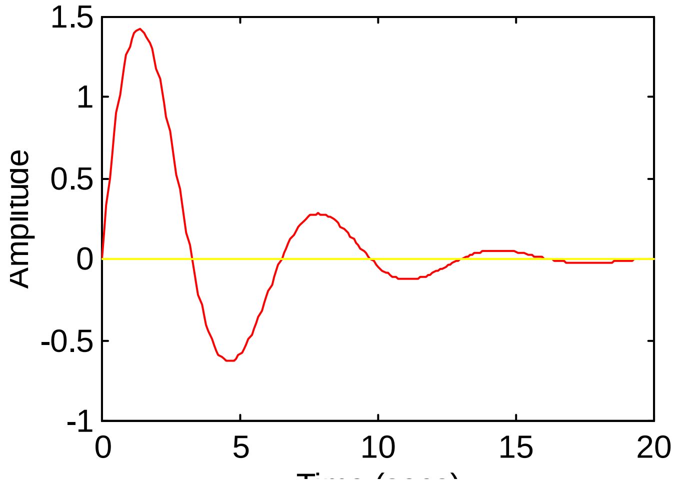
```
>> [n,d]=feedback(n1,d1,n2,d2)
n =
   0  10   1
d =
  10  14   2
```

FdT dei sistemi
serie
parallelo
feedback

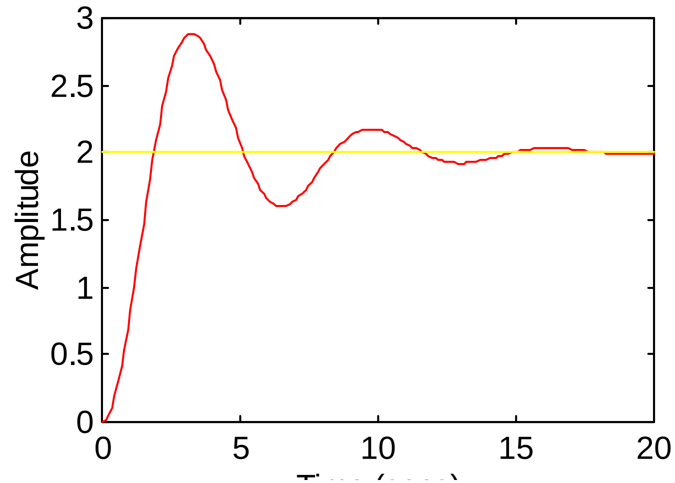
Risposte Canoniche

» num=2; den=[1 0.5 1];

» impulse(num,den)



» step(num,den)



Ingresso qualsiasi

```
» t=[0:.1:40];
```

vettore del tempo

```
» u=zeros(size(t));
» u(10:1:200)=ones(1,191);
```

Inizialmente u è un vettore di zeri, delle dimensioni di t.

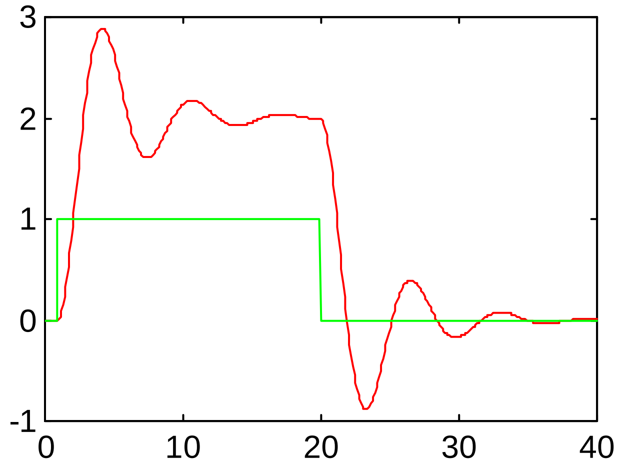
Una parte viene messa a 1

```
» y=lsim(num,den,u,t);

» plot(t,y,'r',t,u,'g')
```

Grafico di u(t) in verde

```
» printsys(num,den,'s')
           2
  ---------------------
  s^2 + 0.5 s + 1
```

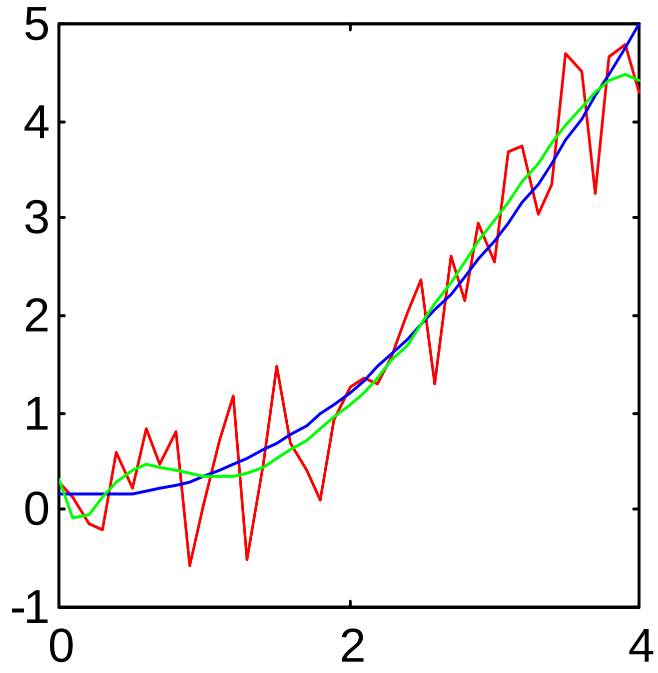
Stampa “in chiaro” della FdT



```
» t=[0:.1:4];
» y= t - sin(t) + 0.2*randn(size(t));
```

grado del polinomio

```
» poly2=polyfit(t,y,2)
poly2 =
   0.3427   -0.1646    0.1707

» poly10=polyfit(t,y,10);
```




```
» plot(t,y,'r',t,polyval(poly2,t),'b',t,polyval(poly10,t),'g')
```

Pendolo Non Lineare

$$J\ddot{\vartheta}+D\dot{\vartheta}+Mgd\sin\vartheta=0 \qquad \begin{cases}\dot{\vartheta}=\omega \\ J\dot{\omega}=-D\omega-Mgd\sin\vartheta\end{cases}$$

Nel file “pend.m”

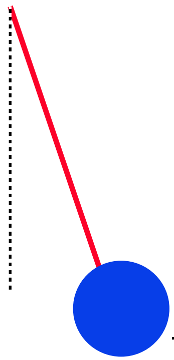
```
function yp=pend(t,y);
% pendolo NL, y(1)=theta;  y(2)=omega
yp(1)=y(2);
yp(2)=-sin(y(1))-0.1*y(2);
end
```

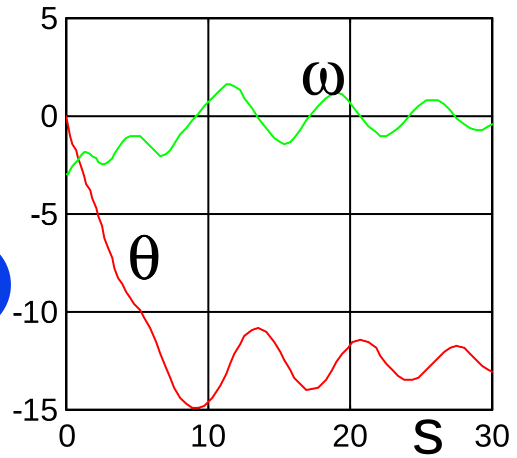
Comandi

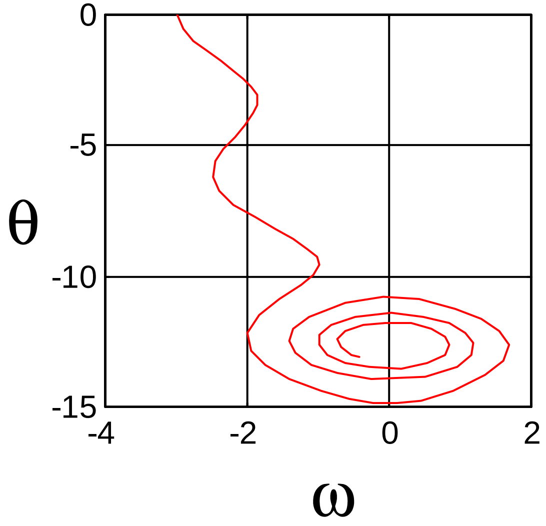
```
[t,y] = ode23('pend',0,30,[0;-3]);

plot(t,y(:,1),'r',t,y(:,2),'g')),grid
plot(y(:,2),y(:,1),'r'),grid
```

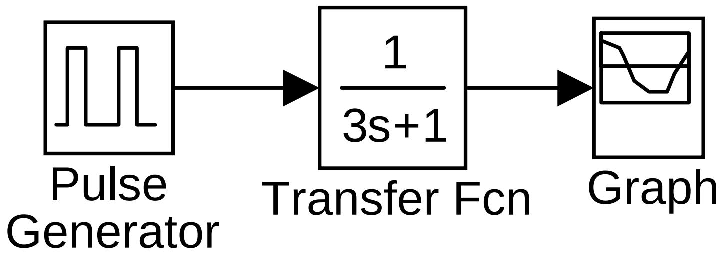
da 0 a 30s

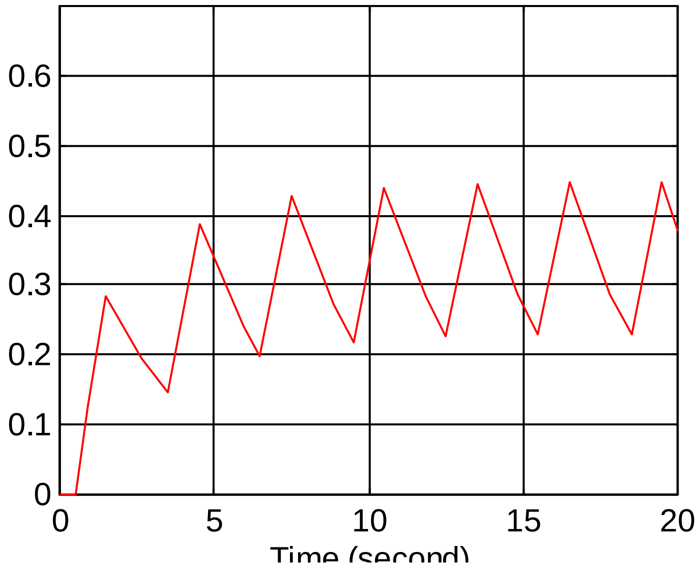
C.I.: $\theta=0$, $\omega=3$








1
3s+1
Pulse
Generator
Transfer Fcn
Graph


0.6
0.5
0.4
0.3
0.2
0.1
0
0
5
10
15
20