

CONTROLLI AUTOMATICI LS
Ingegneria Informatica

INTRODUZIONE A MATLAB

Gianni Borghesan

gianni.borghesan@unibo.it

■ **Analisi** di sistemi:

- SISO (single input - single output)
- MIMO (multiple input – multiple output)
- Lineari e non lineari (Lyapunov)
- Dominio del tempo (continuo e discreto): spazio degli stati
- Dominio delle frequenze (Bode, Nyquist): funzione di trasferimento
- ...

■ **Sintesi** di controllori:

- Controllo PID
- Pole placement
- Controllo ottimo (LQR – LQG)
- ...

- **Control System Toolbox**: fornisce strumenti per la modellazione, l'analisi ed il controllo di sistemi dinamici

- Definizione di variabili e valutazione di espressioni numeriche:

```
>> a=3; b=5;
```

```
>> a*b
```

```
ans = 15
```

- Vettori:

```
>> a = [1 2 3 4 5 6 9 8 7]
```

```
    a =  1  2  3  4  5  6  9  8  7
```

```
>> a = [1:7]
```

```
    a =  1  2  3  4  5  6  9  8  7
```

```
>> a=[1:2:10]
```

```
    a =  1  3  5  7  9
```

- Matrici:

```
>> A = [1 2; 3 4]
```

```
    A =  1  2
```

```
        3  4
```

Matlab presenta un gran numero di funzioni predefinite

- Funzioni Matematiche Predefinite

- `>> sin(pi/3)`
`ans = 0.8660`

- Funzioni "Element Wise" su Vettori:

- `>> a=[0:pi/3:pi]; sin(a)`
`ans = 0 0.8660 0.8660 0.0000`

- `>> a=[1 2 3]; a.*a`
`ans = 1 4 9`

- L'operatore ";" sopprime l'output su video e permette di concatenare i comandi.
- Nel caso non si assegni il risultato di una funzione viene assegnato alla variabile predefinita "ans".

- Vettori
 - Prodotto scalare e vettoriale, somma, sottrazione: $a*b$, $a+b$, $a-b$
 - Vettore trasposto (coniugato): a'
 - Dimensione: $\text{length}(a)$
 - Selezione: $a(1)$, $a(2:2:\text{length}(a))$
- Selezione di parti di matrici:
 - $A(:,2)$: seconda colonna
 - $A(1,2)$: elemento della prima riga / seconda colonna
- Operazioni: $A+A$, $A-A$, $A*A$, A^2
- Matrice trasposta (coniugata): A'
- Matrice inversa: $\text{inv}(A)$
- Determinante: $\text{det}(A)$
- Autovalori: $\text{eig}(A)$
- Rango: $\text{rank}(A)$
- Traccia: $\text{trace}(A)$
- Norma: $\text{norm}(A)$
- Dimensioni: $\text{size}(A)$

- Matrici:
 - Matrice identità $n \times n$: **eye(n)**
 - Matrice di zeri $n \times m$: **zeros(n,m)**
 - Matrice di uni $n \times m$: **ones(n,m)**
 - Matrice $n \times m$ di valori random da 0 a 1: **rand(n,m)**
 - Polinomio caratteristico: **poly()**

- **clear** : per cancellare una variabile (clear all pulisce il workspace)
- **whos**: elenca le variabili definite e ne specifica tipo e dimensione

- Polinomi:

- Definizione di un polinomio

```
>> pol=[1 3 -15 -2 9]
```

```
>> pol =
```

```
1 3 -15 -2 9
```

$$s^4 + 3s^3 - 15s^2 - 2s + 9$$

- Calcolo delle radici:

```
>> roots(pol)
```

```
>> ans =
```

```
-5.5745
```

```
2.5836
```

```
-0.7951
```

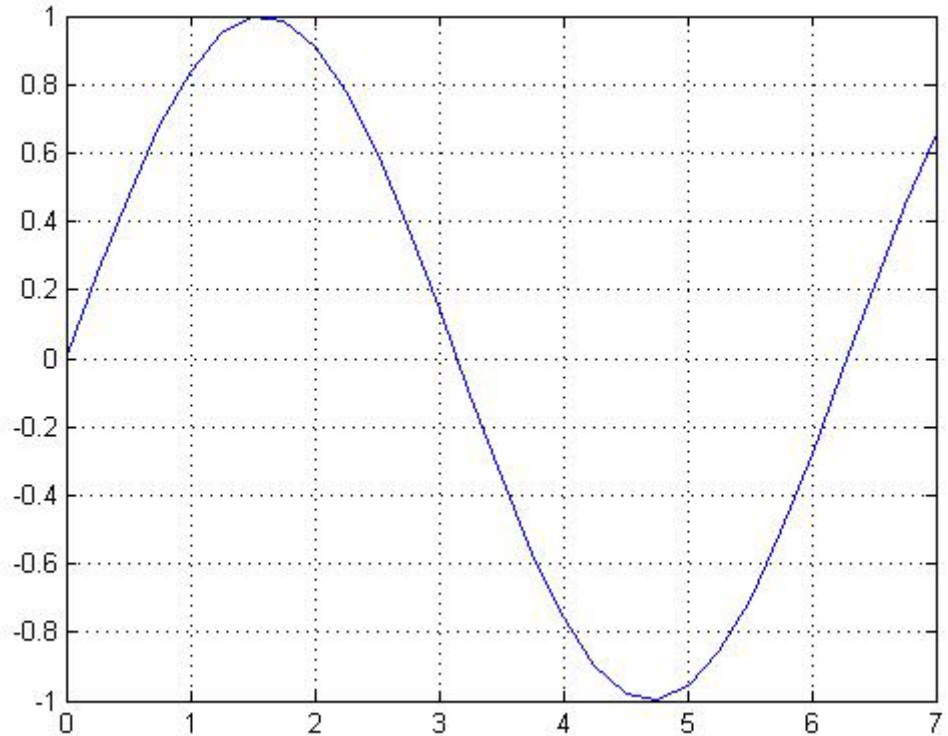
```
0.7860
```

■ Plot:

```
>> t=0:0.25:7;  
>> y = sin(t);  
>> plot(t,y)
```

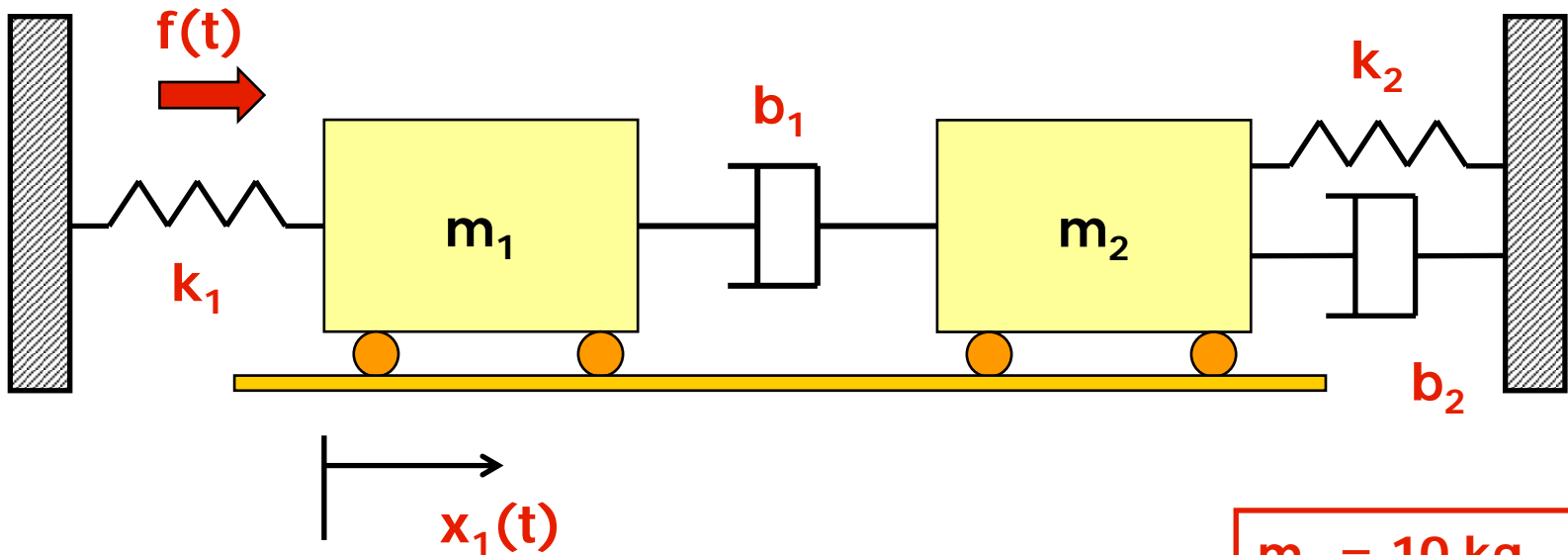
Comandi:

- `axis[Xmin Xmax Ymin Ymax]`
- `grid on / grid off`
- `title, xlabel / ylabel, legend`
- `hold on, hold off`
- `print -depsc 'nomefile.eps'`
- `set(gca,.....)`



- M-files: file contenenti codice Matlab
 - Scripts:
 - Lista di istruzioni
 - Non ha argomenti di input e output
 - Le variabili sono globali
 - Istruzioni: if/then/else, while/end, for...

 - Funzioni:
 - **function[out1,out2,...] = nome_funzione[in1,in2,...]**
 - Hanno variabili di input e output
 - Le variabili sono locali alla funzione
 - Istruzioni: if/then/else, while/end, for...



- Ingresso: $f(t) \rightarrow U(s)$
- Uscita: $x_1(t) \rightarrow Y(s)$

$$G(s) = \frac{Y(s)}{U(s)}$$

$m_1 = 10 \text{ kg}$
$m_2 = 5 \text{ kg}$
$k_1 = 100 \text{ N/m}$
$k_2 = 50 \text{ N/m}$
$b_1 = 10 \text{ Ns/m}$
$b_2 = 20 \text{ Ns/m}$

$$G(s) = \frac{m_2 s^2 + (b_1 + b_2)s + k_2}{m_1 m_2 s^4 + (m_1(b_1 + b_2) + b_1 m_2)s^3 + (b_1 b_2 + k_2 m_1 + k_1 m_2)s^2 + (k_2 b_1 + k_1 b_1 + k_1 b_2)s + k_1 k_2}$$

- $G = \text{tf}(\text{numG}, \text{denG})$ o $G = \text{zpk}(\mathbf{Z}, \mathbf{P}, \mathbf{K})$

- $\text{numG} = [m_2 \quad (b_1 + b_2) \quad k_2]$

$$m_2s^2 + (b_1 + b_2)s + k_2$$

- $\text{denG} = [m_1m_2 \quad (m_1(b_1 + b_2) + b_1m_2) \quad (b_1b_2 + k_2m_1 + k_1m_2) \quad (k_2b_1 + k_1b_1 + k_1b_2) \quad k_1k_2]$

$$m_1m_2s^4 + (m_1(b_1 + b_2) + b_1m_2)s^3 + (b_1b_2 + k_2m_1 + k_1m_2)s^2 + (k_2b_1 + k_1b_1 + k_1b_2)s + k_1k_2$$

- $[\text{numG}, \text{denG}] = \text{tfdata}(G, 'v')$ or $[\mathbf{Z}, \mathbf{P}, \mathbf{K}] = \text{zpkdata}(G, 'v')$

- Risposta all'impulso:

```
>> figure(1)
```

```
>> t = 10; %sec
```

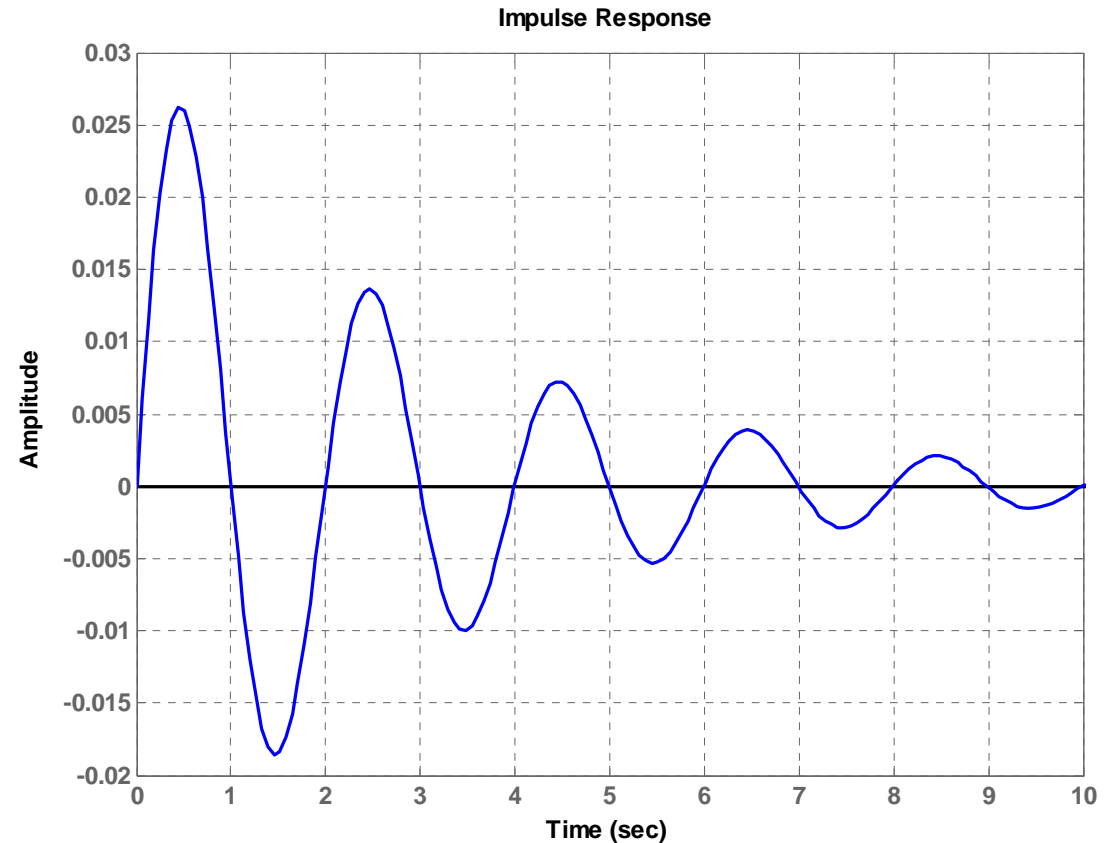
```
>> impulse(G, t)
```

```
>> grid on
```

- Oppure, per salvare la risposta nel w.s.

```
>> [yi,t]=impulse(G, t)
```

```
>> plot (t,yi)
```



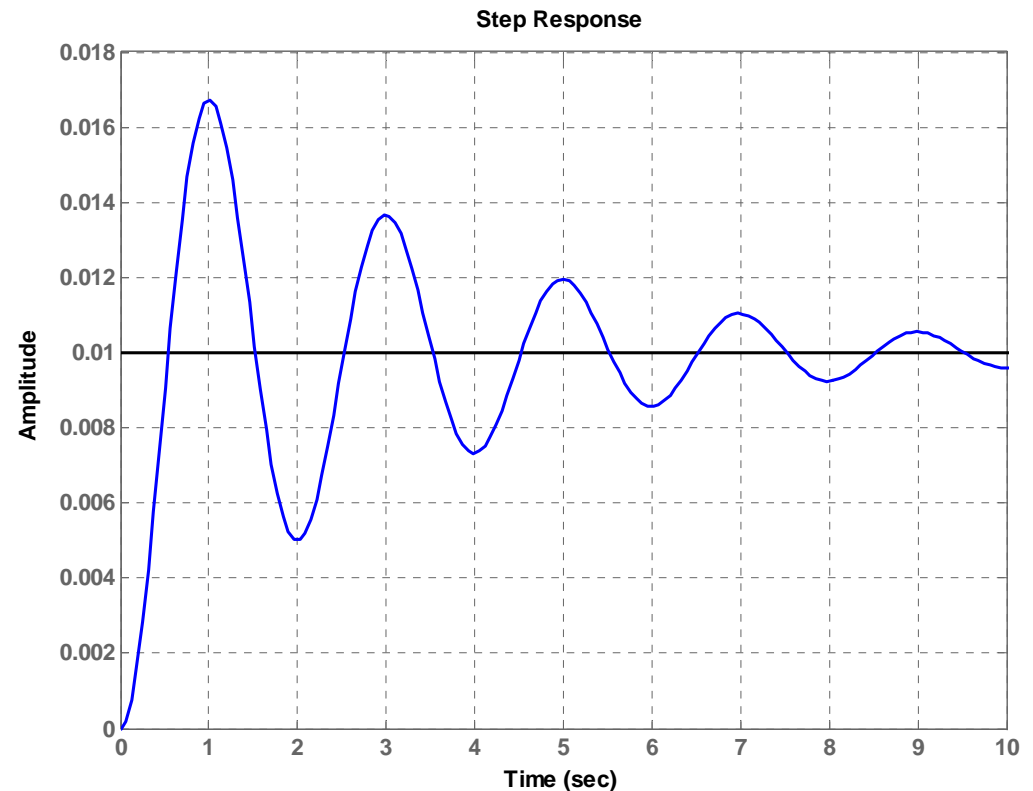
- Risposta al gradino unitario:

- >> figure(2)

- >> t = 10; % sec

- >> **step(G, t)**

- >> grid on



- Risposta ad un segnale generico: $f(t)$

```
>> figure(3)
```

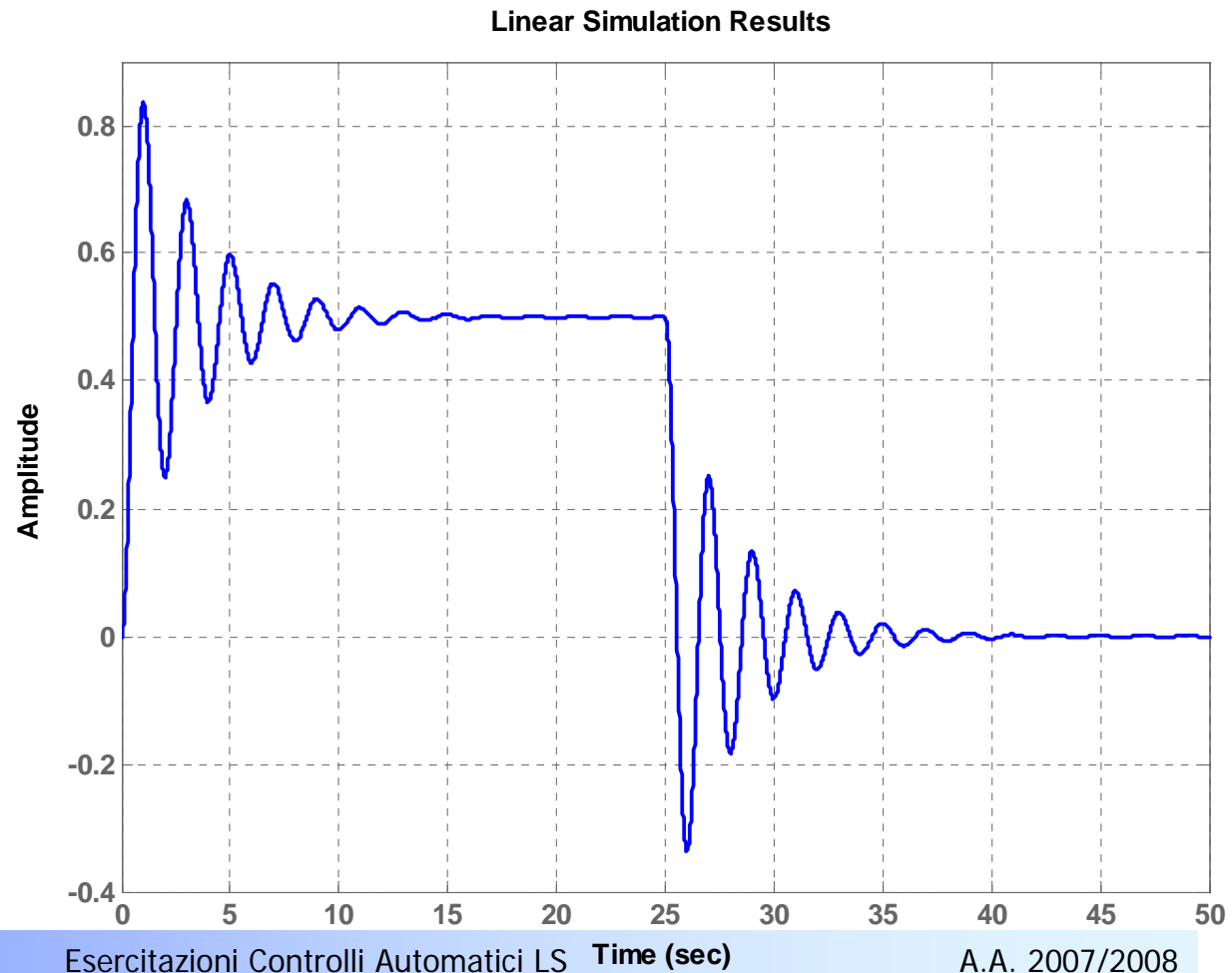
```
>> t = 0:0.01:50; % intervallo di tempo campionato (sec)
```

```
>> f = 50*[ones(1,2500), zeros(1,2500) ones(1,1)]; % segnale di ingresso
```

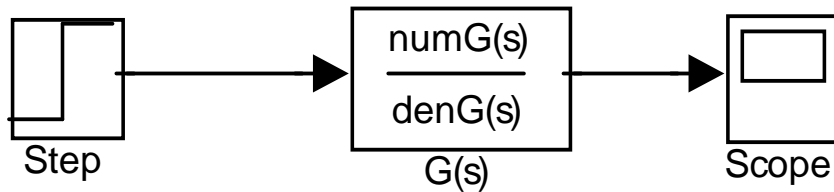
```
>> lsim(G, f, t)
```

```
>> grid on
```

```
>> axis([0 50 -0.4 0.9])
```



■ Simulazione



Function Block Parameters: G(s)

Transfer Fcn

Matrix expression for numerator, vector expression for denominator. Output width equals the number of rows in the numerator. Coefficients are for descending powers of s.

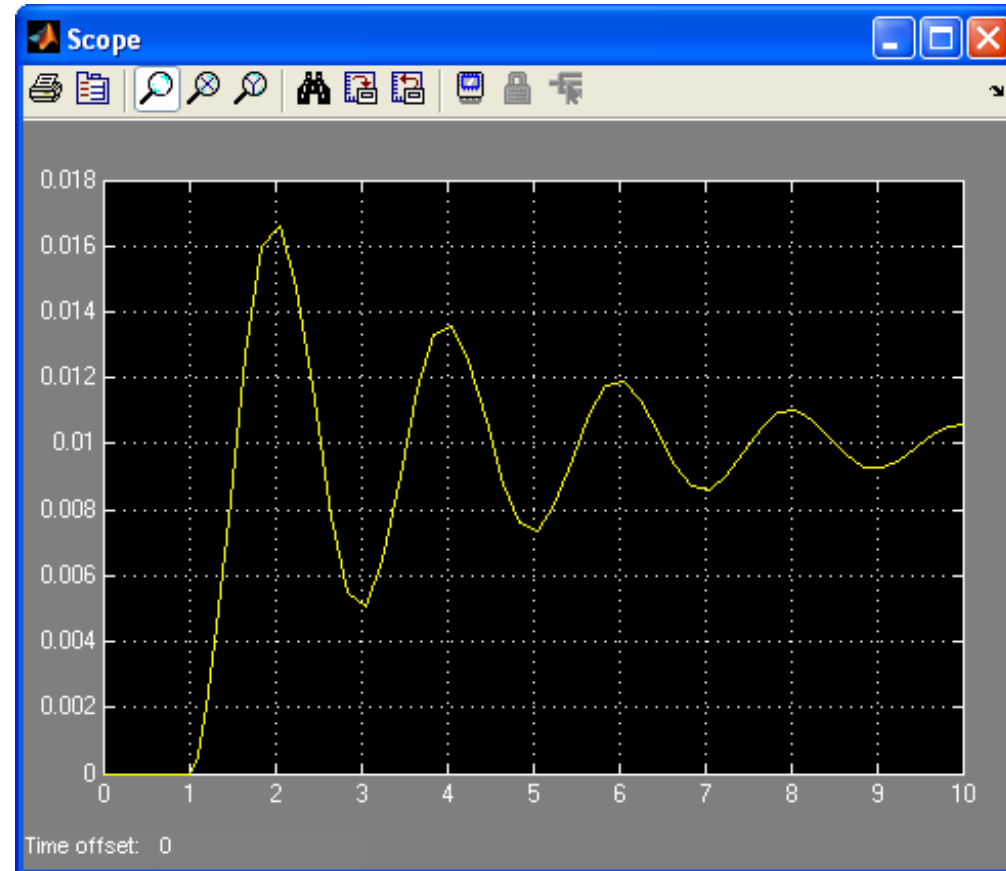
Parameters

Numerator:

Denominator:

Absolute tolerance:

OK Cancel Help Apply



- Conversione tempo continuo / tempo discreto:

```
>> G_discreta = c2d(G, Tsample, 'metodo')
```

- Conversione tempo discreto / tempo continuo:

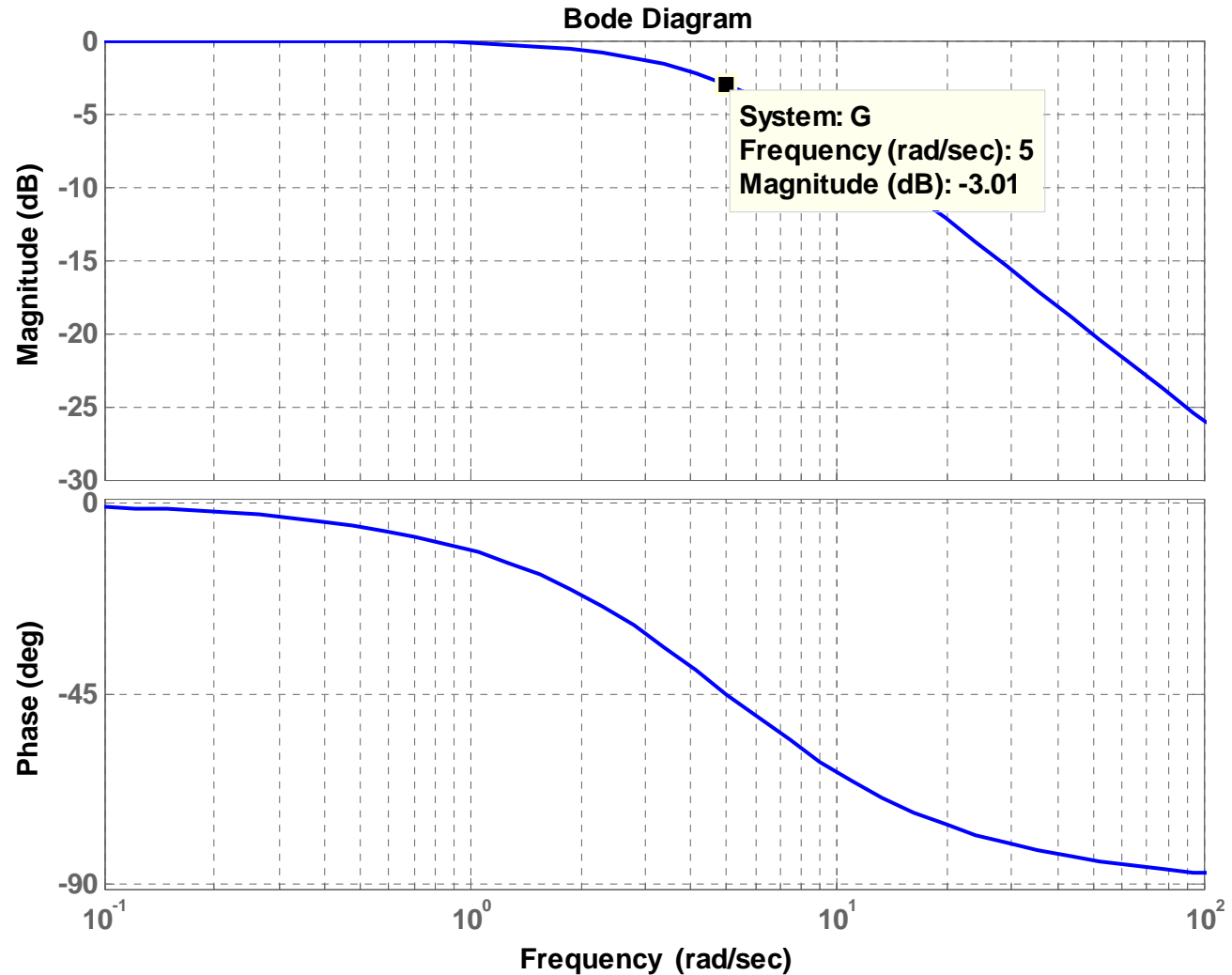
```
>> G = d2c(G_discreta, 'metodo')
```


- Metodo:

- **zoh**: zero-order hold
- **tustin**: approssimazione bilineare (o di Tustin)
- **prewarp**: approssimazione di Tustin con precompensazione frequenziale
- **matched**: corrispondenza poli/zeri (solo per sistemi SISO)

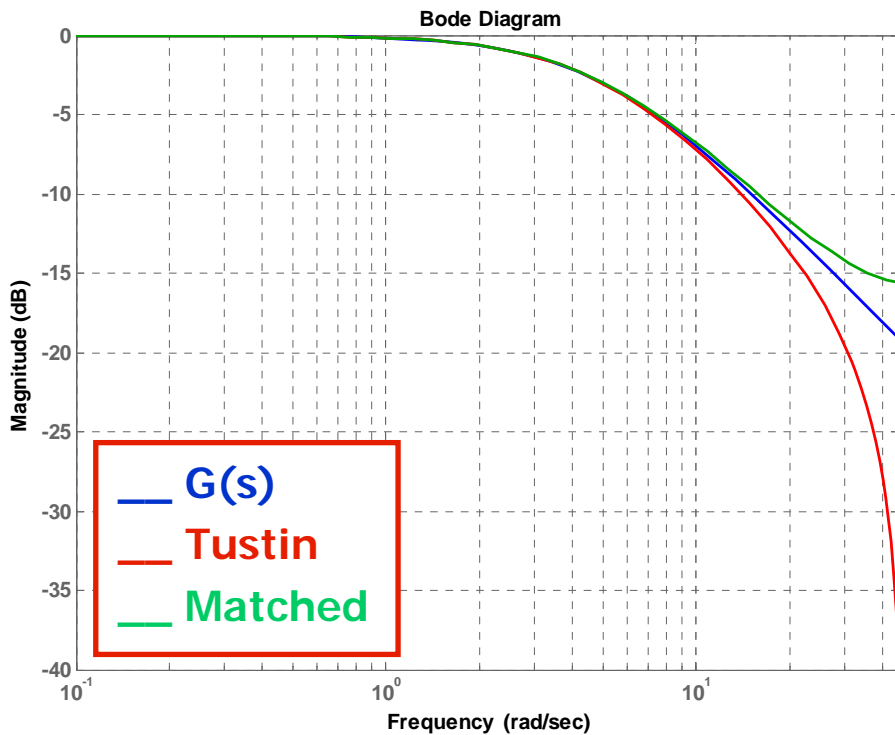
- >> **bode(G)**, grid on $G(s) = \frac{5}{s+5}$

$$\omega_c = 5 \text{ rad/sec}$$



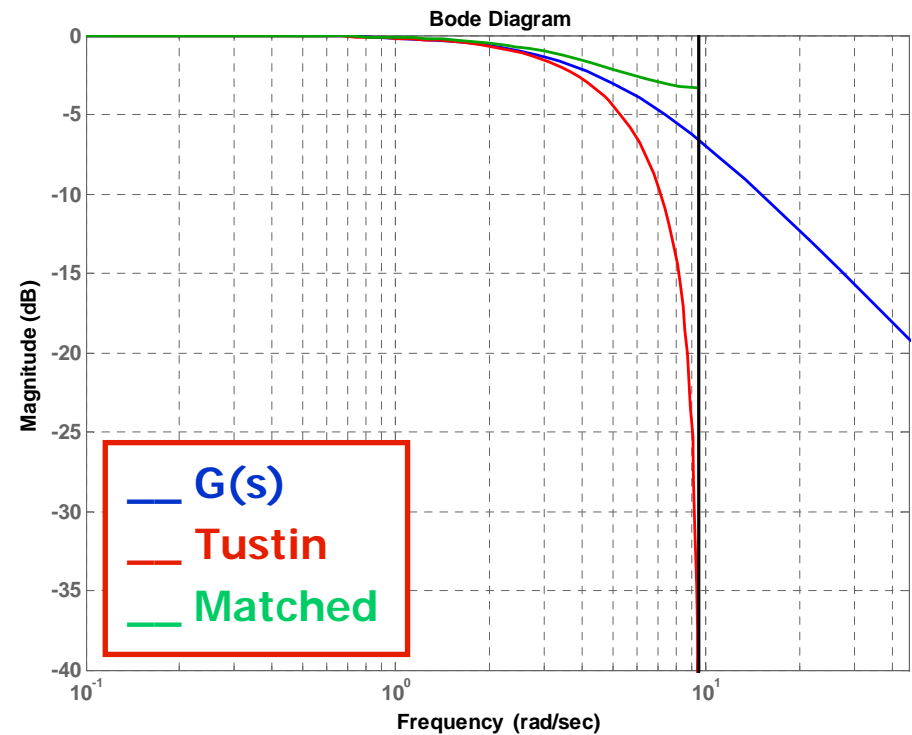
- **Regola pratica** $\alpha\omega_c \leq \omega_s \leq 10\alpha\omega_c$, con $5 \leq \alpha \leq 10$ $\omega_s = \frac{2\pi}{T}$
- $\omega_c = 5$, $\alpha = 5 \Rightarrow 25 \leq \omega_s \leq 250$  $4 \leq \frac{1}{T} \leq 40$
- Conversione tempo continuo / tempo discreto:
 - >> $T = 1/15$; %sec $\omega_s \approx 100\text{rad/sec}$
 - >> **G_T1 = c2d (G, T, 'tustin');**
 - >> **G_M1 = c2d (G, T, 'matched');**
 - >> **bodemag(G,'b', G_T1,'r--', G_M1, 'g--')**

 - >> $T = 1/3$; %sec $\omega_s \approx 20\text{rad/sec}$
 - >> **G_T2 = c2d (G, T, 'tustin');**
 - >> **G_M2 = c2d (G, T, 'matched');**
 - >> **bodemag(G,'b', G_T2,'r--', G_M2, 'g--')**



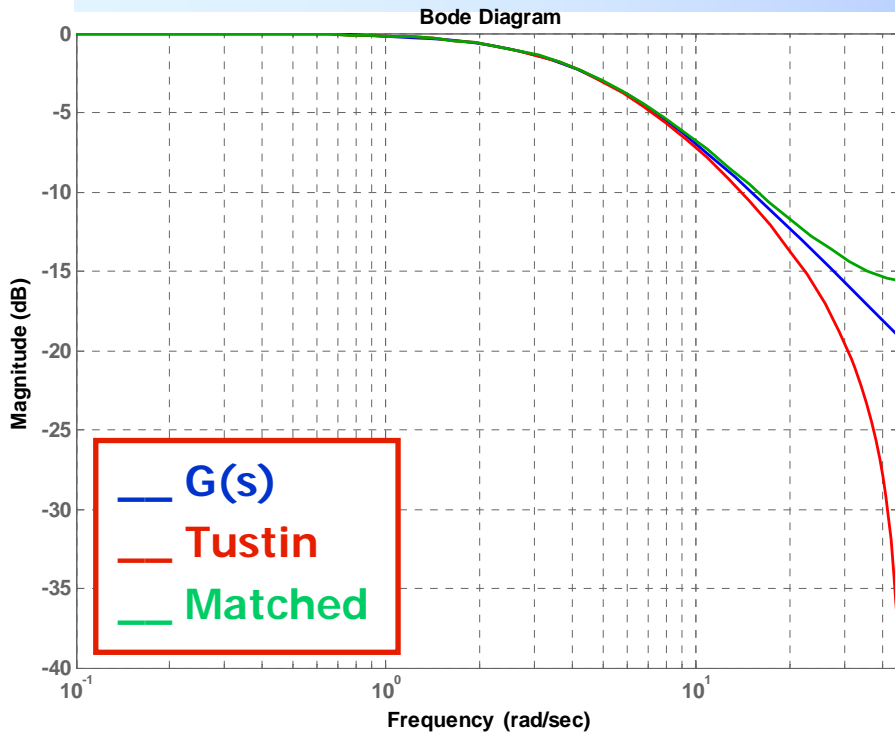
$$T = 1/15 \text{ sec}$$

$$\omega_s \approx 100 \text{ rad/sec}$$



$$T = 1/3 \text{ sec}$$

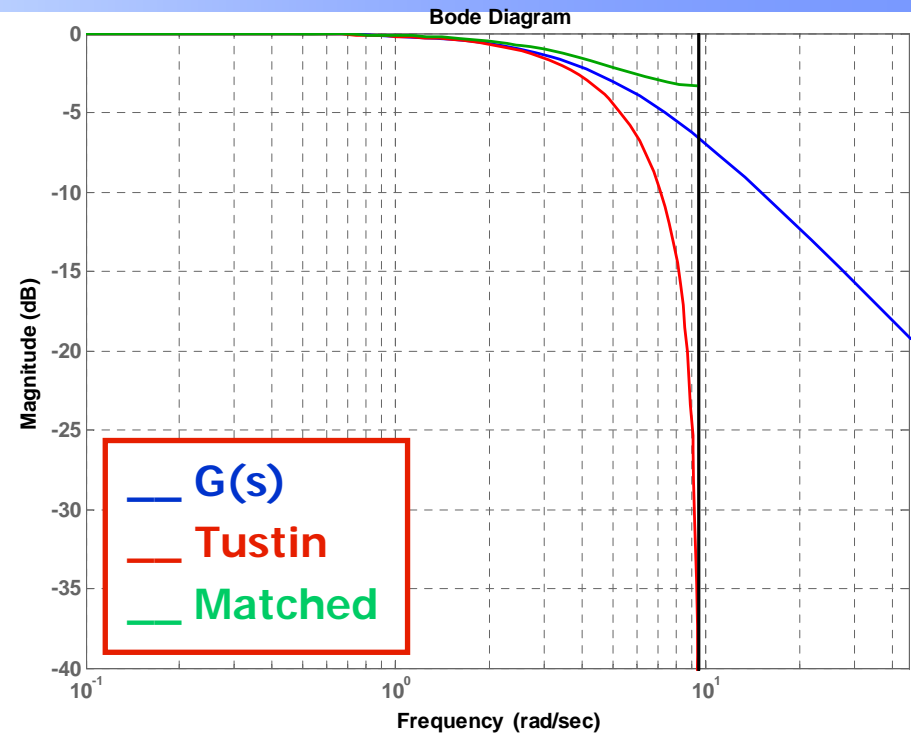
$$\omega_s \approx 20 \text{ rad/sec}$$



$$T = 1/15 \text{ sec} \quad \omega_s \approx 100 \text{ rad/sec}$$

$$G_{T1}(z) = \frac{0.1429z + 0.1429}{z - 0.7143}$$

$$G_{M1}(z) = \frac{0.2835}{z - 0.7165}$$



$$T = 1/3 \text{ sec} \quad \omega_s \approx 20 \text{ rad/sec}$$

$$G_{T2}(z) = \frac{0.4545z + 0.4545}{z - 0.09091}$$

$$G_{M2}(z) = \frac{0.8111}{z - 0.1889}$$

CONTROLLI AUTOMATICI LS

Introduzione a Matlab **FINE**

Gianni Borghesan

gianni.borghesan@unibo.it