

# Matlab e Simulink per i sistemi di controllo

**Analisi e simulazione in ambiente  
Matlab**

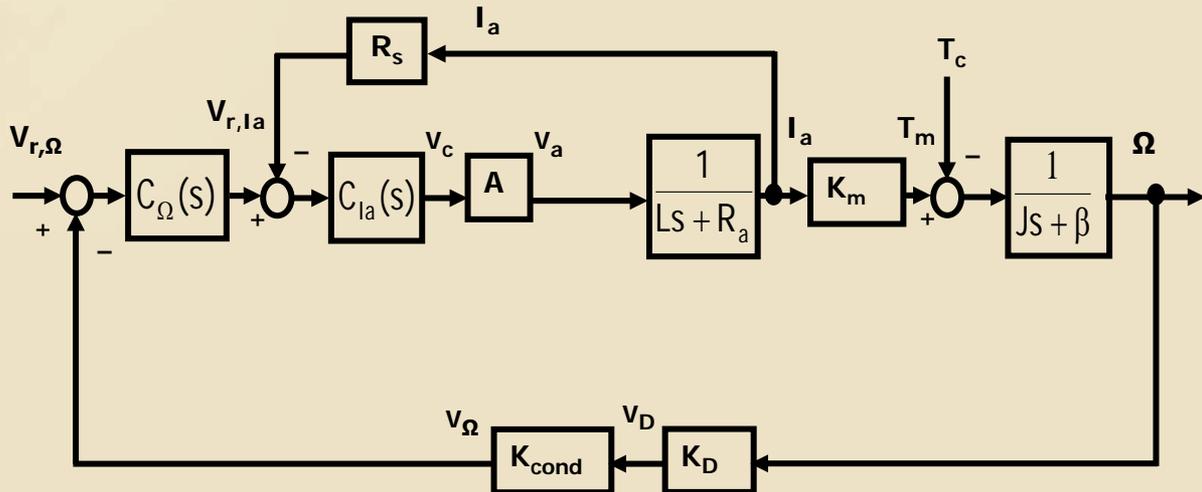


## Introduzione (1/2)

- L'utilizzo del linguaggio **MATLAB**® permette di realizzare facilmente i principali passi necessari per l'analisi ed il progetto di sistemi di controllo:
  - Manipolazione di funzioni di trasferimento e calcolo delle loro principali caratteristiche (singolarità, guadagno, ecc.)
  - Analisi del comportamento in frequenza di un sistema mediante il tracciamento di diagrammi di Bode, di Nyquist e di Nichols della sua fdt
  - Simulazione della risposta di un sistema ad un ingresso assegnato

# Un esempio applicativo

- L'utilizzo di Matlab per l'analisi e la simulazione di un sistema di controllo sarà illustrato nel caso di un **servomotore in c.c.** rappresentato dal seguente schema a blocchi

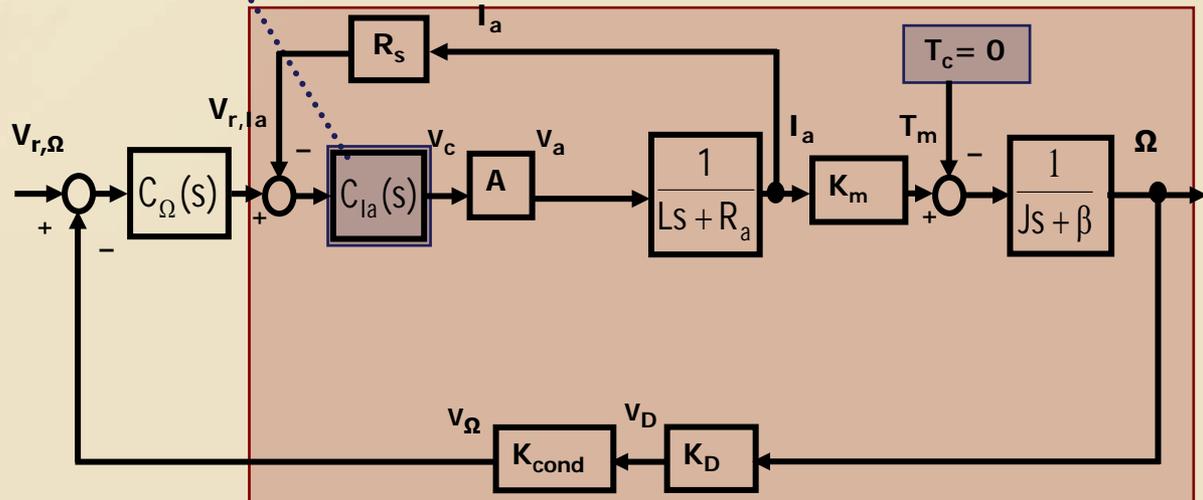


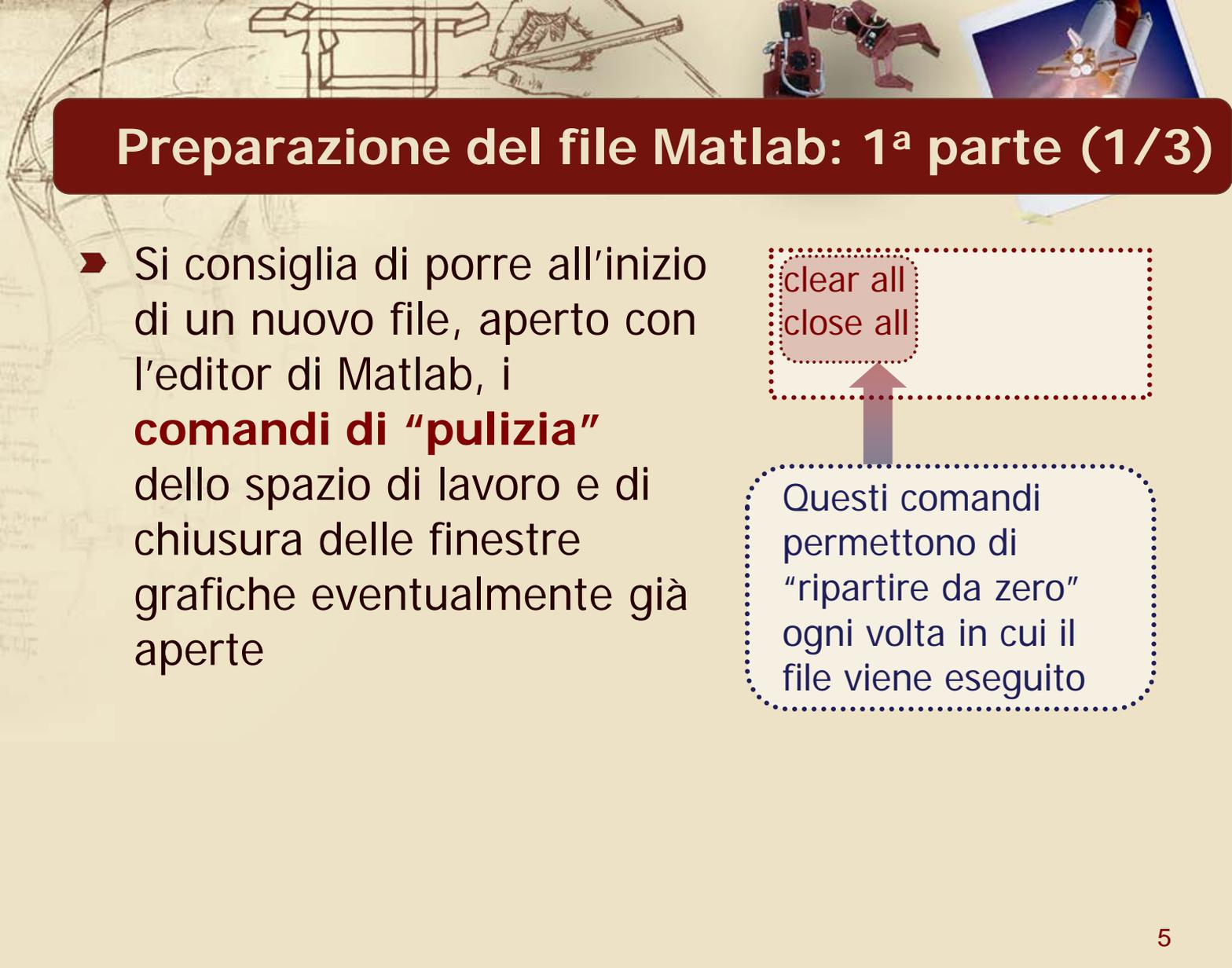
# Calcolo della fdt del motore

- **Primo obiettivo:** calcolo della fdt del motore (incluso anello di corrente) per  $T_c = 0$

$$C_{I_a}(s) = \frac{K}{s}$$

$$F(s) = \frac{V_{\Omega}(s)}{V_{r,la}(s)}$$





## Preparazione del file Matlab: 1<sup>a</sup> parte (1/3)

- Si consiglia di porre all'inizio di un nuovo file, aperto con l'editor di Matlab, i **comandi di "pulizia"** dello spazio di lavoro e di chiusura delle finestre grafiche eventualmente già aperte

```
clear all  
close all
```

Questi comandi permettono di "ripartire da zero" ogni volta in cui il file viene eseguito

# Preparazione del file Matlab: 1<sup>a</sup> parte (1/3)

- Si consiglia di porre all'inizio di un nuovo file, aperto con l'editor di Matlab, i **comandi di "pulizia"** dello spazio di lavoro e di chiusura delle finestre grafiche eventualmente già aperte
- Si definisce la **variabile complessa s** per un'agevole definizione delle fdt di sistemi o sottosistemi

```
clear all  
close all  
s=tf('s');
```

Il comando tf permette di definire e/o di calcolare la fdt di un sistema LTI secondo diverse modalità

# Preparazione del file Matlab: 1<sup>a</sup> parte (2/3)

- Si assegnano ai parametri del sistema i corrispettivi **valori numerici**. (nelle appropriate unità di misura) e si definisce  $C_{Ia}(s)$

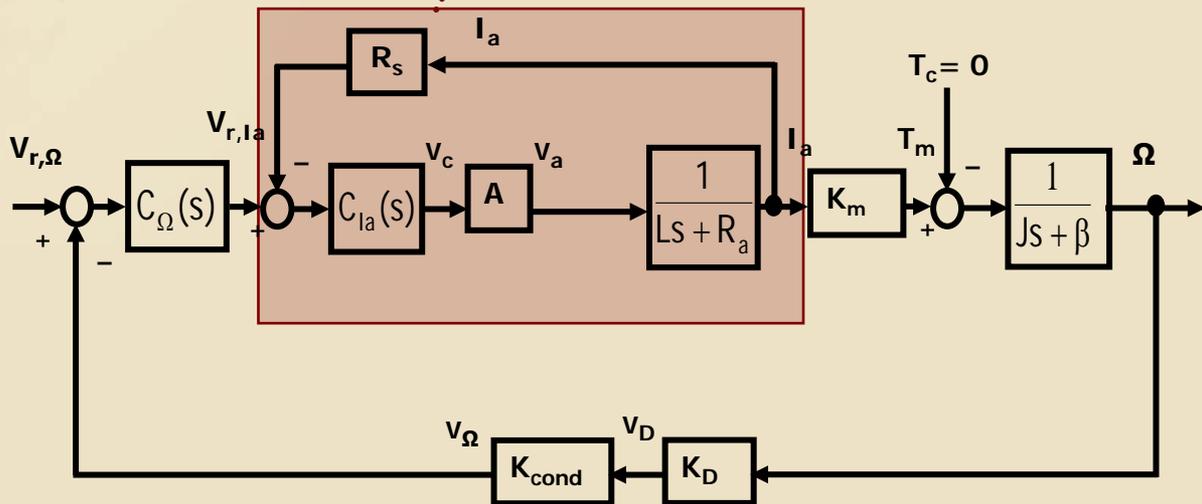
```
clear all
close all
s=tf('s');
Ra=6;
L=3.24e-3;
Km=0.0535;
J=20e-6;
beta=14e-6;
KD=0.0285;
Kcond=0.67;
Rs=7.525;
A=2.925;
K=1000;
Cia=K/s;
```

# Preparazione del file Matlab: 1<sup>a</sup> parte (3/3)

- Si calcola la fdt fra  $V_{r,la}(s)$  e  $I_a(s)$  (indicata come  $F_{r,la}(s)$ ) con il comando **feedback**

1° argomento: fdt del ramo diretto

$F_{r,la} = \text{feedback}(C_{la} * A / (L * s + R_a), R_s);$

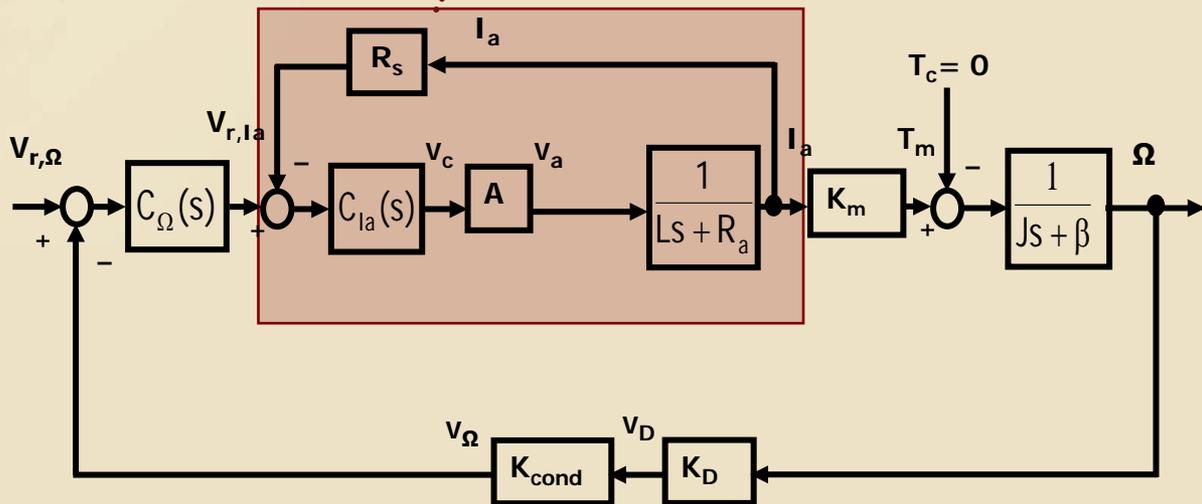


# Preparazione del file Matlab: 1<sup>a</sup> parte (3/3)

- Si calcola la fdt fra  $V_{r,la}(s)$  e  $I_a(s)$  (indicata come  $F_{r,la}(s)$ ) con il comando **feedback**

2° argomento: fdt della retroazione

$F_{r,la} = \text{feedback}(C_{la} * A / (L * s + R_a), R_s)$

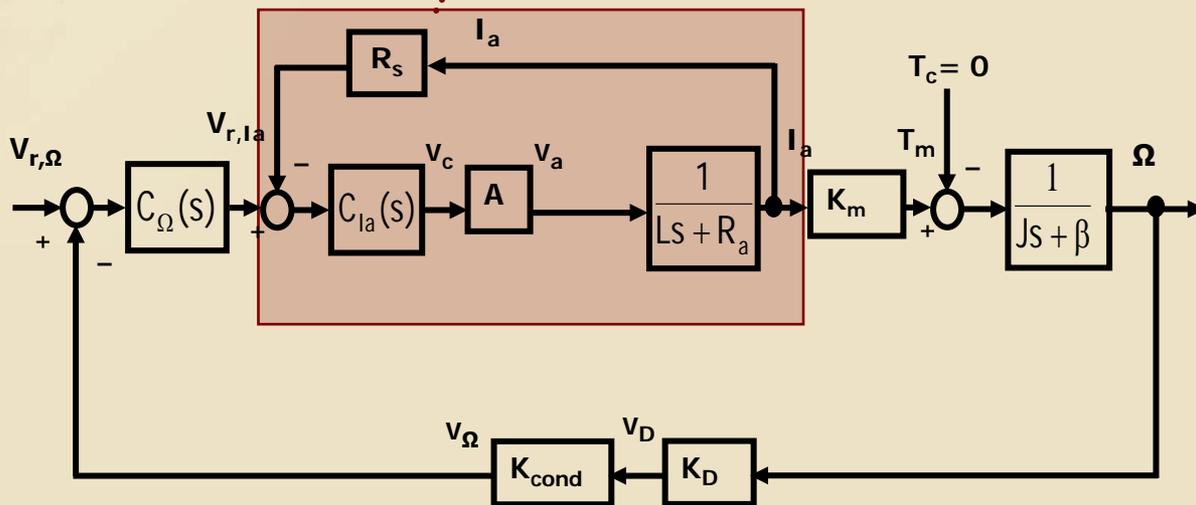


# Preparazione del file Matlab: 1<sup>a</sup> parte (3/3)

- Si calcola la fdt fra  $V_{r,la}(s)$  e  $I_a(s)$  (indicata come  $F_{r,la}(s)$ ) con il comando **feedback**

La retroazione è assunta automaticamente negativa

$F_{r,la} = \text{feedback}(C_{la} * A / (L * s + R_a), R_s);$



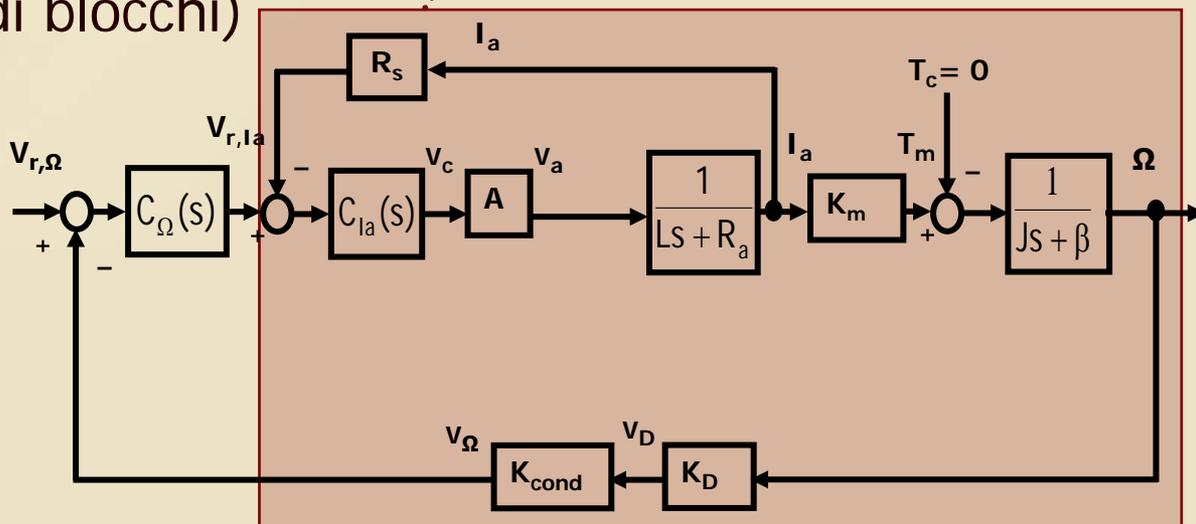
# Preparazione del file Matlab: 1<sup>a</sup> parte (3/3)

- Si calcola la fdt fra  $V_{r,la}(s)$  e  $I_a(s)$  (indicata come  $F_{r,la}(s)$ ) con il comando **feedback**

Per visualizzare il risultato non si mette il ;

- Si calcola la  $F(s)$  cercata (cascata di blocchi)

$F_{r,la} = \text{feedback}(C_{Ia} * A / (L * s + R_a), R_s)$ ;  
 $F = F_{r,la} * K_m / (J * s + \beta) * K_D * K_{cond}$



# Fdt del motore e schema a blocchi risultante

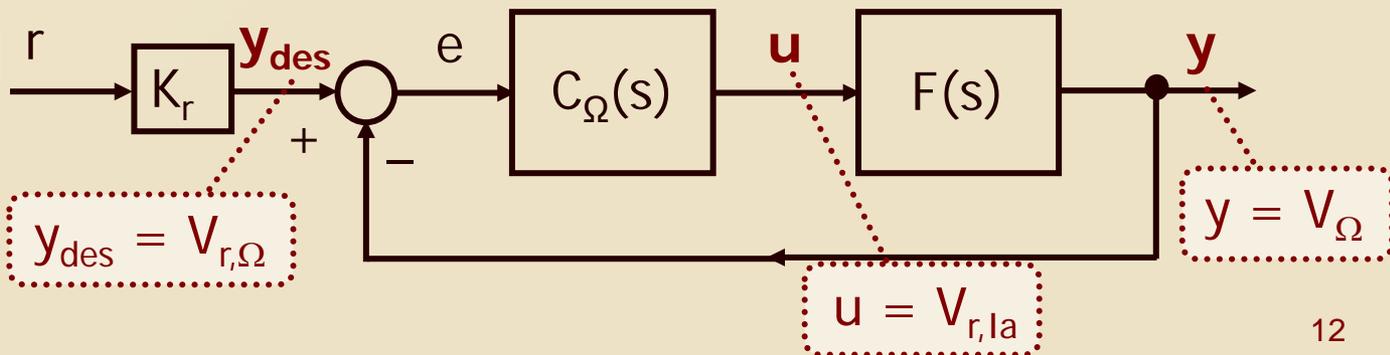
- L'esecuzione della prima parte del file Matlab così preparata dà come risultato:

Transfer function:

2.988

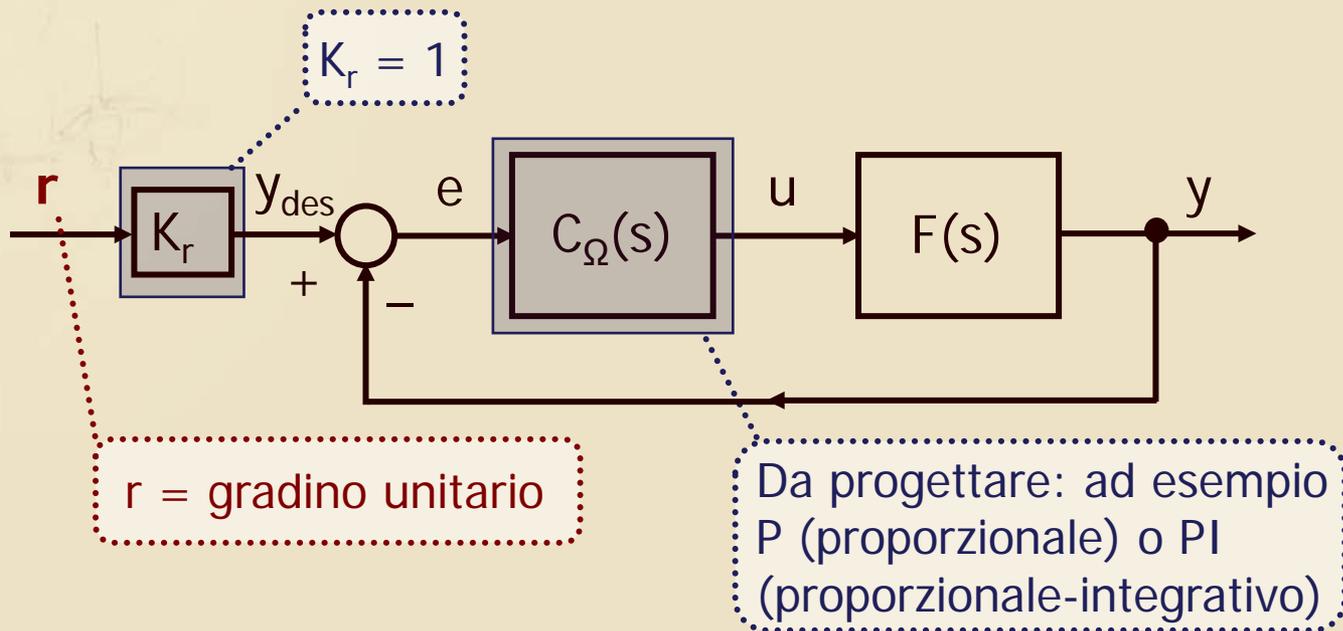
$6.48e-008 s^3 + 0.00012 s^2 + 0.4403 s + 0.3081$

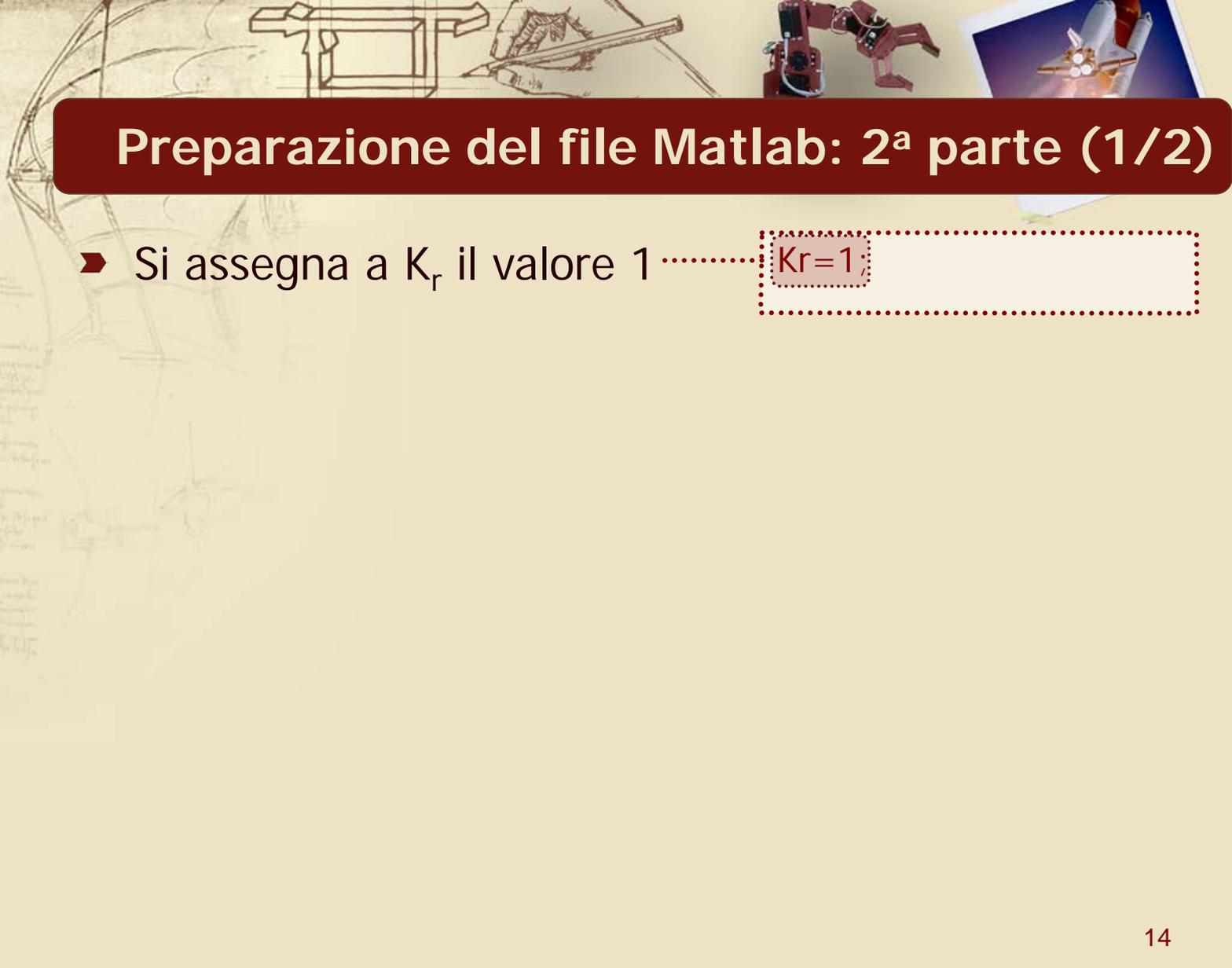
- Per  $T_c = 0$  lo schema a blocchi diventa pertanto:



# Simulazione del sistema ad anello chiuso

- **Secondo obiettivo:** simulazione della risposta del sistema ad anello chiuso ad un riferimento a gradino unitario, per diversi controllori  $C_{\Omega}(s)$





## Preparazione del file Matlab: 2<sup>a</sup> parte (1/2)

- Si assegna a  $K_r$  il valore 1

```
Kr=1;
```

## Preparazione del file Matlab: 2<sup>a</sup> parte (1/2)

- Si assegna a  $K_r$  il valore 1
- Si definisce la fdt del controllore:  $K_p$  (proporzionale)

```
Kr=1;  
Kp=0.4;  
C_omega1=Kp;
```

## Preparazione del file Matlab: 2<sup>a</sup> parte (1/2)

- Si assegna a  $K_r$  il valore 1
- Si definisce la fdt del controllore:  $K_p$  (proporzionale)  
 $K_p + K_i/s$  (proporzionale-integrativa)

```
Kr=1;  
Kp=0.4;  
C_omega1=Kp;  
Ki=2;  
C_omega2=Kp+Ki/s;
```

## Preparazione del file Matlab: 2<sup>a</sup> parte (1/2)

- Si assegna a  $K_r$  il valore 1
- Si definisce la fdt del controllore:  $K_p$  (proporzionale)  
 $K_p + K_i/s$  (proporzionale-integrativa)
- Si calcola la fdt ad anello chiuso nei due casi

```
Kr=1;  
Kp=0.4;  
C_omega1=Kp;  
Ki=2;  
C_omega2=Kp+Ki/s;
```

```
W1=Kr*feedback(C_omega1*F,1);  
W2=Kr*feedback(C_omega2*F,1);
```

## Preparazione del file Matlab: 2ª parte (2/2)

- Si applica il gradino unitario con il comando **step**

```
Kr=1;  
Kp=0.4;  
C_omega1=Kp;  
Ki=2;  
C_omega2=Kp+Ki/s;
```

```
W1=Kr*feedback(C_omega1*F,1);  
W2=Kr*feedback(C_omega2*F,1);  
step(W1,5)  
hold on  
step(W2,5)  
hold off
```

2° argomento: istante  
finale della simulazione  
a partire da  $t = 0$

## Preparazione del file Matlab: 2<sup>a</sup> parte (2/2)

- Si applica il gradino unitario con il comando **step**

```
Kr=1;  
Kp=0.4;  
C_omega1=Kp;  
Ki=2;  
C_omega2=Kp+Ki/s;
```

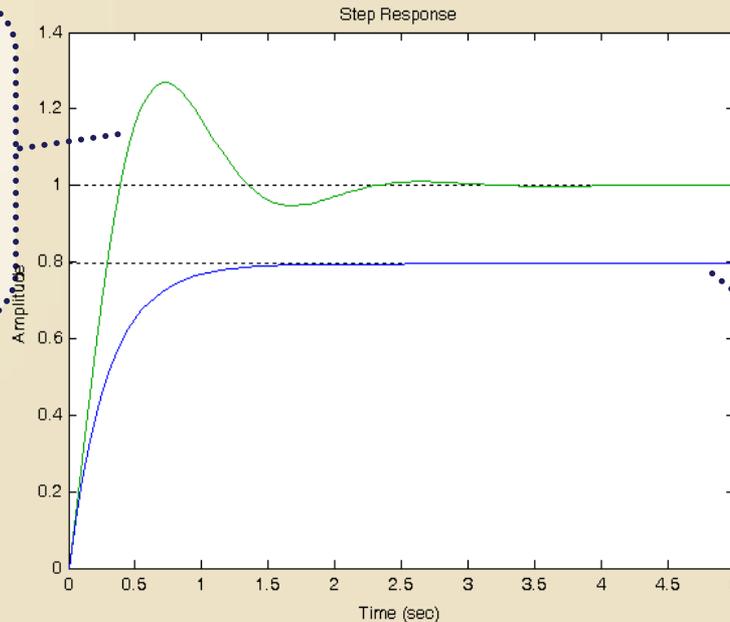
```
W1=Kr*feedback(C_omega1*F,1);  
W2=Kr*feedback(C_omega2*F,1);  
step(W1,5)  
hold on  
step(W2,5)  
hold off
```

Tutti i grafici vengono riportati nella medesima finestra per un più agevole confronto

# Risultato della simulazione

- L'esecuzione della seconda parte del file Matlab così preparata effettua la simulazione desiderata:

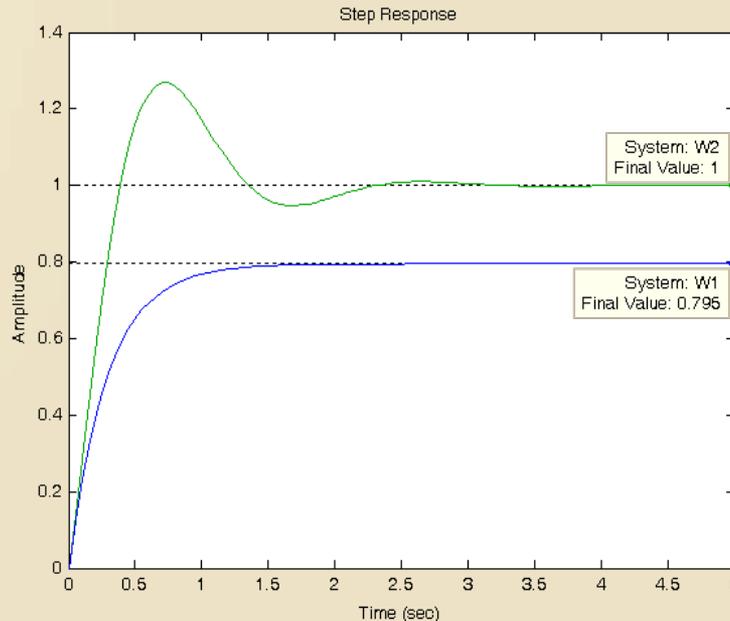
Risposta al gradino unitario con  
 $C_{\Omega}(s) = K_p + K_i/s$



Risposta al gradino unitario con  
 $C_{\Omega}(s) = K_p$

# Analisi della risposta del sistema (1/3)

- Valutazione del valore della **risposta in regime permanente** e calcolo dell'errore di inseguimento finale

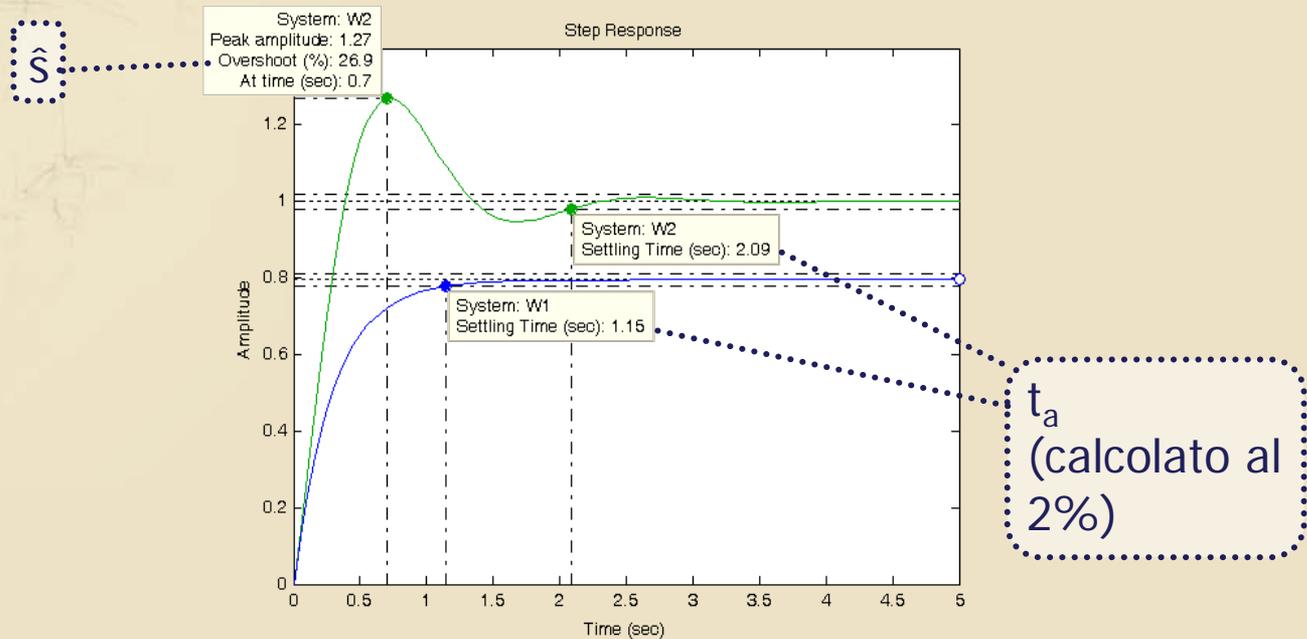


$$e_{\infty} = 0$$

$$e_{\infty} = 0.205$$

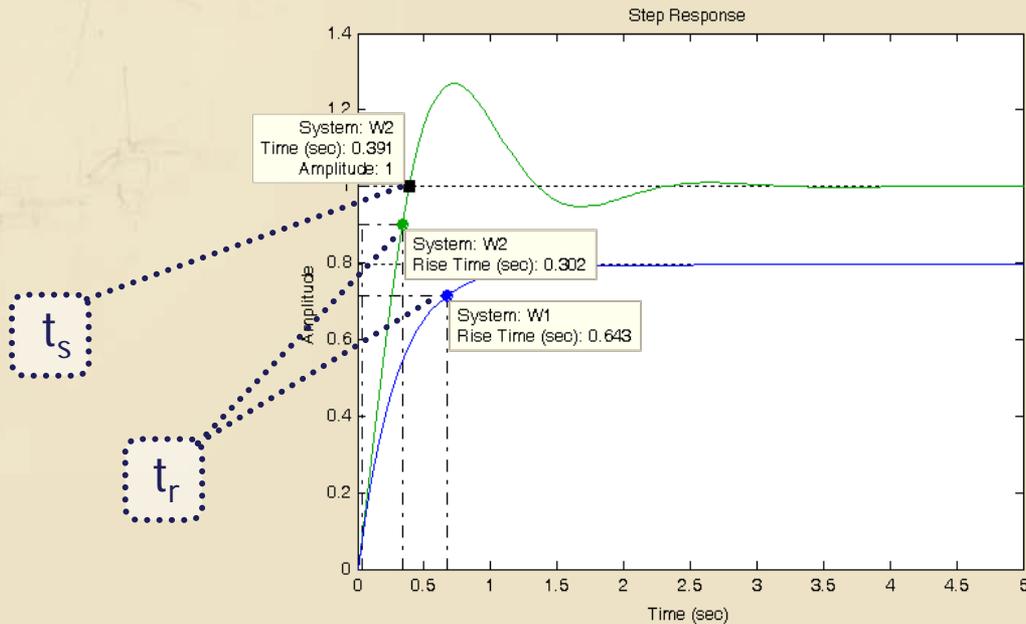
# Analisi della risposta del sistema (2/3)

- Valutazione della **sovraelongazione massima** e del **tempo di assestamento**



# Analisi della risposta del sistema (3/3)

- Valutazione del **tempo di salita** (secondo le definizioni date per  $t_r$  e  $t_s$ )



## Poli del sistema ad anello chiuso (1/2)

- Il diverso comportamento del sistema ad anello chiuso con i due controllori adottati può essere giustificato calcolando il valore dei **poli** della funzione  $W(s)$  nei due casi:

damp(W1)  
damp(W2)



Il comando damp determina i poli della fdt nella forma "parte reale + parte immaginaria" e ne fornisce anche pulsazione naturale e fattore di smorzamento

## Poli del sistema ad anello chiuso (2/2)

Eigenvalue	Damping	Freq. (rad/s)
-3.42e+000	1.00e+000	3.42e+000
-9.25e+002 + 2.44e+003i	3.55e-001	2.61e+003
-9.25e+002 - 2.44e+003i	3.55e-001	2.61e+003

damp(W1)  
damp(W2)

## Poli del sistema ad anello chiuso (2/2)

Eigenvalue	Damping	Freq. (rad/s)
-3.42e+000	1.00e+000	3.42e+000
-9.25e+002 + 2.44e+003i	3.55e-001	2.61e+003
-9.25e+002 - 2.44e+003i	3.55e-001	2.61e+003

damp(W1)  
damp(W2)

Eigenvalue	Damping	Freq. (rad/s)
-1.71e+000 + 3.27e+000i	4.63e-001	3.69e+000
-1.71e+000 - 3.27e+000i	4.63e-001	3.69e+000
-9.25e+002 + 2.44e+003i	3.55e-001	2.61e+003
-9.25e+002 - 2.44e+003i	3.55e-001	2.61e+003

# Riconoscimento dei poli dominanti

Eigenvalue	Damping	Freq. (rad/s)
-3.42e+000	1.00e+000	3.42e+000
-9.25e+002 + 2.44e+003i	3.55e-001	2.61e+003
-9.25e+002 - 2.44e+003i	3.55e-001	2.61e+003

Eigenvalue	Damping	Freq. (rad/s)
-1.71e+000 + 3.27e+000i	4.63e-001	3.69e+000
-1.71e+000 - 3.27e+000i	4.63e-001	3.69e+000
-9.25e+002 + 2.44e+003i	3.55e-001	2.61e+003
-9.25e+002 - 2.44e+003i	3.55e-001	2.61e+003

**Poli dominanti**

# Riconoscimento dei poli dominanti

$W_1(s)$  ha un polo dominante reale

**Poli dominanti**

Eigenvalue	Damping	Freq. (rad/s)
-3.42e+000	1.00e+000	3.42e+000
-9.25e+002 + 2.44e+003i	3.55e-001	2.61e+003
-9.25e+002 - 2.44e+003i	3.55e-001	2.61e+003

Eigenvalue	Damping	Freq. (rad/s)
-1.71e+000 + 3.27e+000i	4.63e-001	3.69e+000
-1.71e+000 - 3.27e+000i	4.63e-001	3.69e+000
-9.25e+002 + 2.44e+003i	3.55e-001	2.61e+003
-9.25e+002 - 2.44e+003i	3.55e-001	2.61e+003

# Riconoscimento dei poli dominanti

$W_1(s)$  ha un polo dominante reale

Eigenvalue	Damping	Freq. (rad/s)
-3.42e+000	1.00e+000	3.42e+000
-9.25e+002 + 2.44e+003i	3.55e-001	2.61e+003
-9.25e+002 - 2.44e+003i	3.55e-001	2.61e+003

**Poli dominanti**

$$\zeta = 0.463$$

$W_2(s)$  ha una coppia di poli dominanti complessi coniugati

Eigenvalue	Damping	Freq. (rad/s)
-1.71e+000 + 3.27e+000i	4.63e-001	3.69e+000
-1.71e+000 - 3.27e+000i	4.63e-001	3.69e+000
-9.25e+002 + 2.44e+003i	3.55e-001	2.61e+003
-9.25e+002 - 2.44e+003i	3.55e-001	2.61e+003