

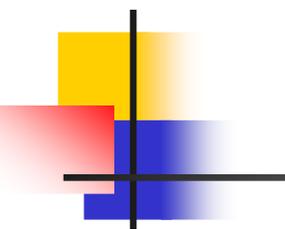
Introduzione a Matlab/Simulink

Robotica Industriale

Prof. P. Rocco

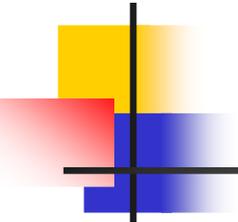
a.a.2003/2004

Ing. M. Gritti e Ing. L. Bascetta



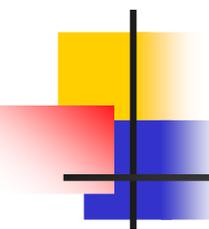
Contenuti

- Introduzione a Matlab
- Presentazione Control System Toolbox
- Introduzione a Simulink
- Esempi



Contenuti

- **Introduzione a Matlab**
 - **Variabili, matrici e vettori e polinomi**
 - **Tracciamento di grafici**
 - **Librerie e Toolbox**
- Presentazione Control System Toolbox
- Introduzione a Simulink
- Esempi



Variabili

- Definizione di variabili
 - » `a=2; b=4;`
 - Non occorre dichiarare preliminarmente le variabili
- Le variabili vengono memorizzate nel Workspace
 - `whos` mostra la lista delle variabili
 - `clear var` elimina la variabile `var` dal workspace
 - `clear all` elimina tutte le variabili del workspace
- Il ``;'` al termine dell'istruzione sopprime la visualizzazione dell'output

Matrici

- Definizione di una matrice

$$A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \quad \gg \quad A = [1, 2; 3, 4];$$

- Selezione degli elementi di una matrice

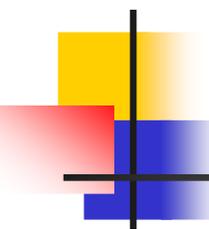
» $A(1, 2)$

ans =

2

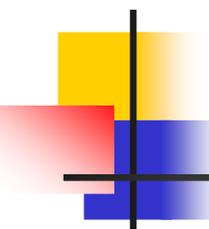
- Per selezionare righe e colonne si utilizza la wildcard `' : '`

- Selezione della prima riga di A $\Rightarrow A(1, :)$
- Selezione della prima colonna di A $\Rightarrow A(:, 1)$
- Selezione di un intervallo $\Rightarrow A(1, 1:2)$



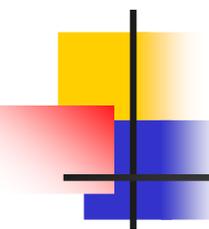
Vettori

- I vettori vengono utilizzati per rappresentare segnali o per la tracciatura di grafici
 - Sintassi
 - » `v = val_iniziale:passo:val_finale;`
 - Esempio
 - » `v=0:100;` % Se il passo è 1 si può omettere
 - » `v=0:0.01:1;`
- Operazioni su vettori
 - » `a + b;` `a - b;`
 - » `a * b` (Errore!!); `a * b' ;`
 - » `a .* b` % Prodotto elemento per elemento
 - » `a.^2` % Quadrato elemento per elemento



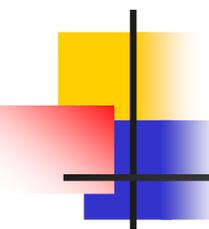
Polinomi

- Si rappresentano come vettori
 - Esempio: $P(s) = s^2 + 2s + 1$
 - » `P = [1 2 1];`
 - Calcolo delle radici
 - » `roots(P)`
 - Costruzione di un polinomio con radici assegnate
 - » `R = poly([-1 -2])` %Ritorna un vettore



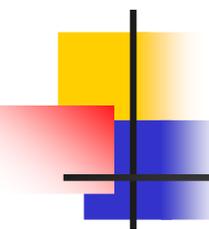
Rappresentazioni grafiche

- `plot(x,y)` : traccia il grafico dei punti che hanno come ascisse gli elementi di `x` e come ordinate gli elementi di `y`
 - Esempio: tracciatura di un'esponenziale
 - » `x=0:0.05:5;`
 - » `y=exp(x);`
 - » `plot(x,y)`



Librerie di funzioni

- Matlab mette a disposizione numerose librerie di funzioni
 - Funzioni generiche (es. `sin`, `abs`, `real`, `eig`, `det`, `roots`)
 - Funzioni specifiche (es. `bode`, `step`, `nyquist`)
- Per ottenere un aiuto si utilizza il comando `help`
 - `help` fornisce l'elenco delle funzioni per libreria
 - `help libreria` fornisce una breve descrizione delle funzioni di una certa libreria
 - `help funzione` fornisce una dettagliata descrizione di una funzione

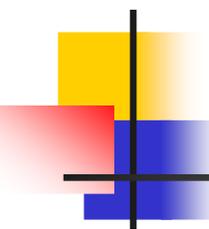


Contenuti

- Introduzione a Matlab
- **Presentazione Control System Toolbox**
 - **Rappresentazione e analisi di un sistema LTI**
- Introduzione a Simulink
- Esempi

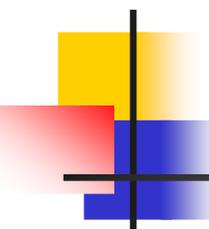
Descrizione di un sistema LTI

- Un sistema LTI può essere specificato:
 - Attraverso una rappresentazione nello spazio con una quaterna di matrici (A,B,C,D)
 - » $A = [-2 \quad -0.5; \quad 4 \quad 0]; \quad B = [1 \quad 0]'$;
 - $C = [-1 \quad 0.25]; \quad d = 0;$
 - » `sys1 = ss(A,B,C,d)`
 - Attraverso una funzione di trasferimento con una coppia di polinomi N(s) e D(s)
 - » $N = [-1 \quad 1]; \quad D = [1 \quad 2 \quad 2];$
 - » `sys2 = tf(N,D)`
 - Le funzioni `ss(sys2)` e `tf(sys1)` permettono la conversione tra le due rappresentazioni



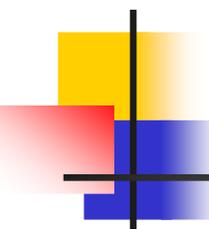
Analisi di un sistema LTI

- Risposta allo scalino
 - » `step(sys)`
- Diagrammi di Bode (solo del modulo)
 - » `bode(sys)` (» `bodemag(sys)`)
- Pulsazione critica e margine di fase
 - » `margin(sys)`
- Diagramma di Nyquist
 - » `nyquist(sys)`
- Mappa poli-zeri
 - » `pzmap(sys)`



Contenuti

- Introduzione a Matlab
- Presentazione Control System Toolbox
- **Introduzione a Simulink**
 - **Introduzione alla simulazione**
 - **Cenni ai metodi di integrazione numerica**
 - **Scelta dei parametri di simulazione**
- Esempi

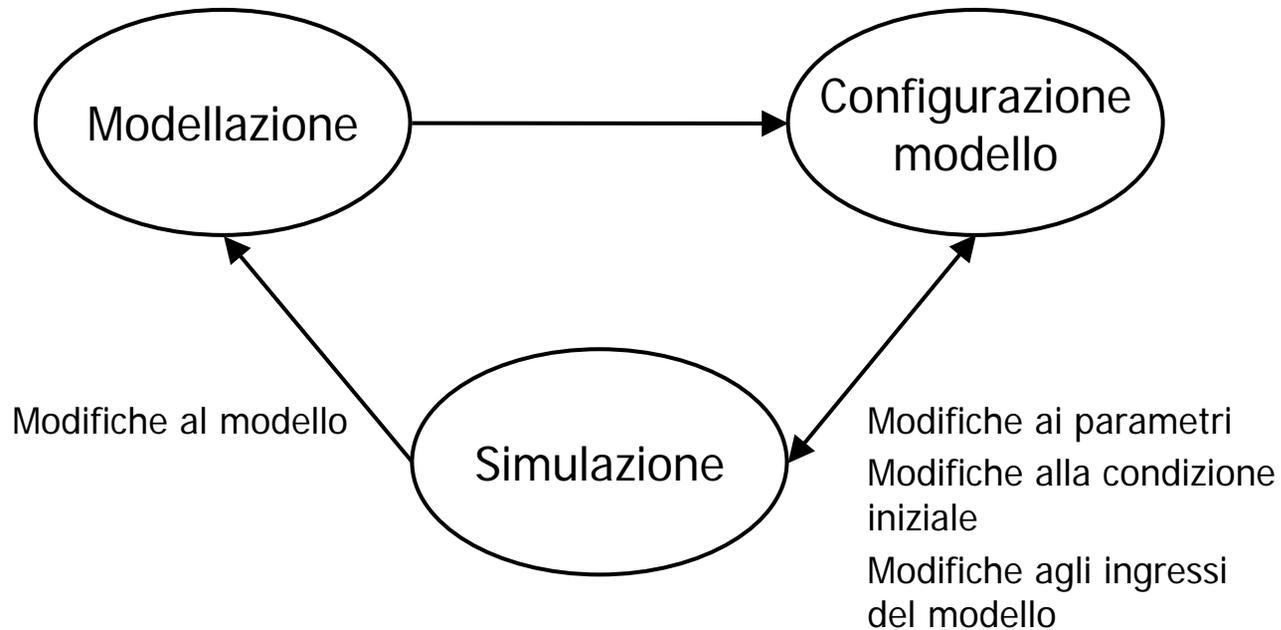


Fasi della simulazione

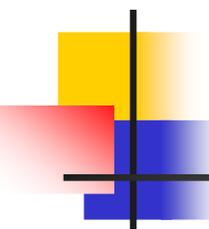
- Scrittura del modello matematico
- Programmazione del modello
 - Con un editor grafico (es. Simulink) o con un linguaggio orientato al calcolo (es. Matlab)
 - Assegnamento dei parametri del modello
- Simulazione (in senso stretto)
 - Calcolo dello stato di regime (condizioni iniziali)
 - Assegnamento dei parametri di simulazione
 - Scelta degli ingressi durante il transitorio
 - Avvio della simulazione
- Analisi dei risultati

Ciclo di una simulazione

- Il ciclo di simulazione procede iterativamente

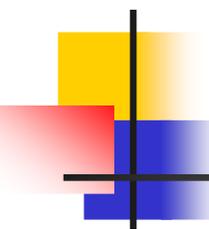


- Modifica dei parametri di simulazione (metodo di integrazione, passo di integrazione, tolleranza, ...)
- Modifica del modello



Introduzione a Simulink (1)

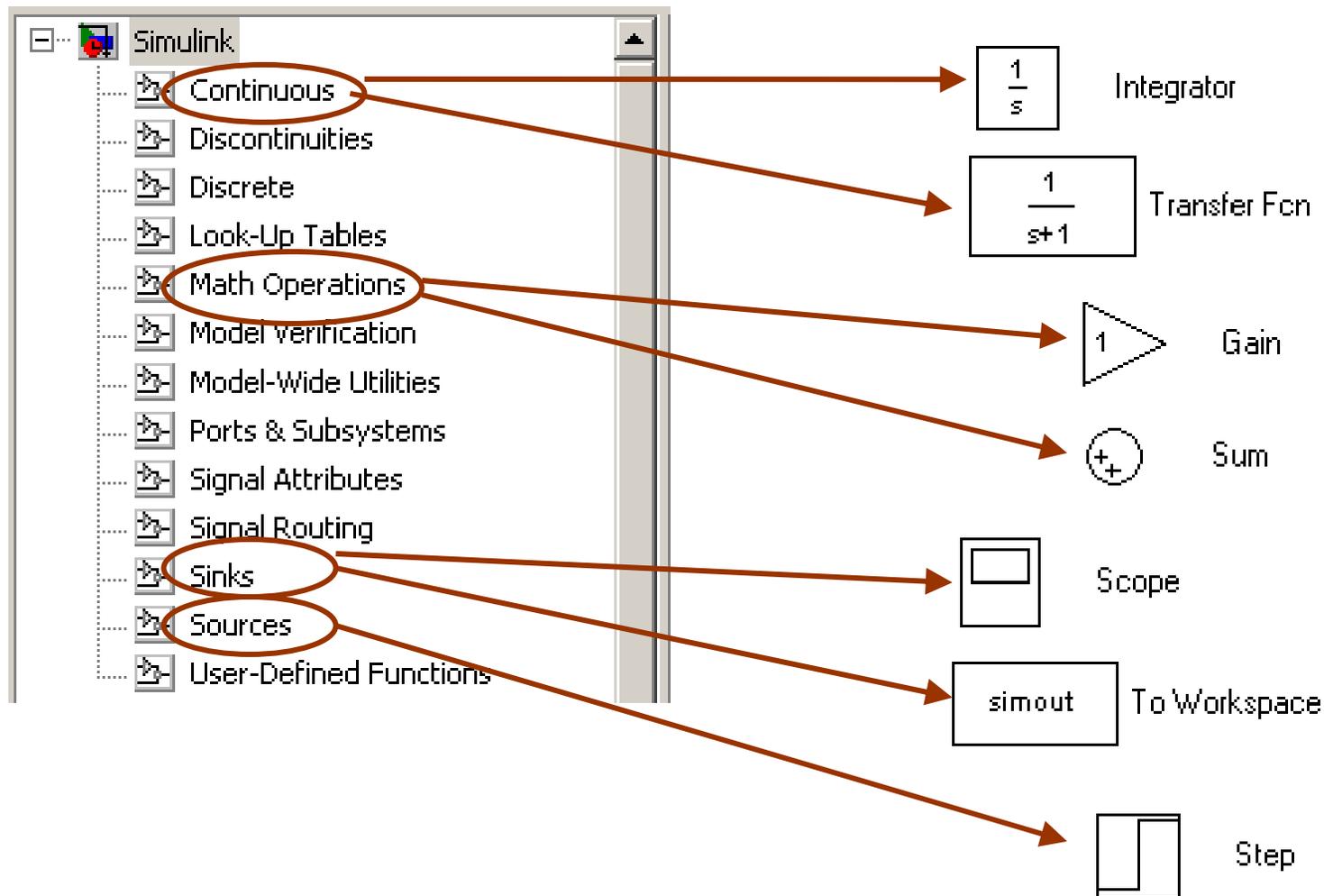
- *Simulink* è un toolbox di MATLAB per la simulazione dei sistemi dinamici
 - Lineari e non lineari
 - Continui, discreti o misti
- Il modello viene costruito graficamente, assemblando blocchi elementari
- I blocchi base sono raggruppati in librerie specializzate, a seconda delle funzioni svolte
- Simulink viene avviato digitando `simulink` al prompt di Matlab



Integrazione Matlab - Simulink

- Simulink e Matlab interagiscono attraverso il *workspace*
 - I parametri del modello possono essere specificati all'interno dei blocchi Simulink tramite variabili simboliche, il cui valore è memorizzato nel workspace di Matlab
 - I risultati della simulazione possono essere memorizzati nel workspace con un blocco di tipo **ToWorkspace**
- Suggerimenti
 - Creare un file `parametri_modello.m` contenente i parametri da caricare prima della simulazione
 - Tracciare i grafici dei transitori con i comandi di Matlab

I blocchi più importanti



Costruzione di un semplice modello

- Simulare la risposta a scalino del sistema

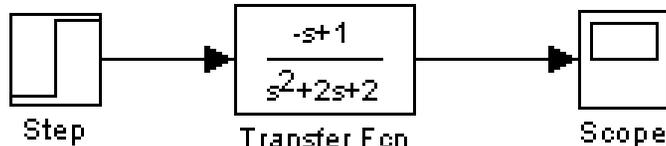
$$G(s) = \frac{-s + 1}{s^2 + 2s + 2}$$

Parameters

Step time:

Initial value:

Final value:



Parameters

Numerator:

Denominator:

Simulation time

Start time: Stop time:

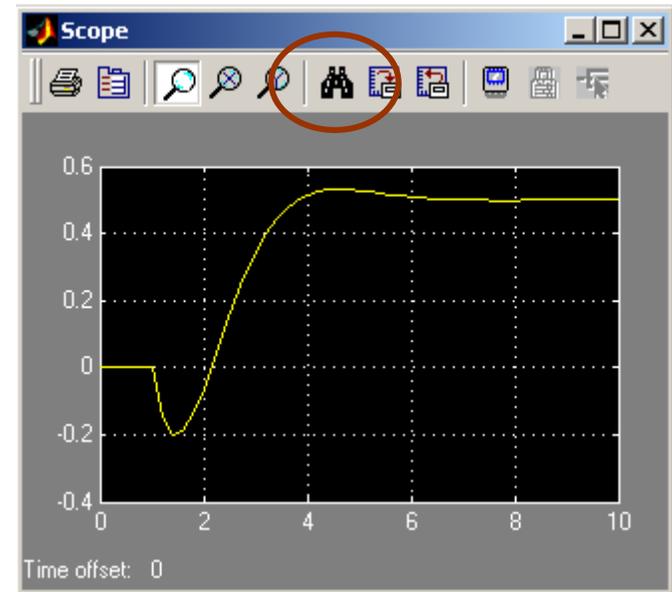
Solver options

Type:

Max step size: Relative tolerance:

Min step size: Absolute tolerance:

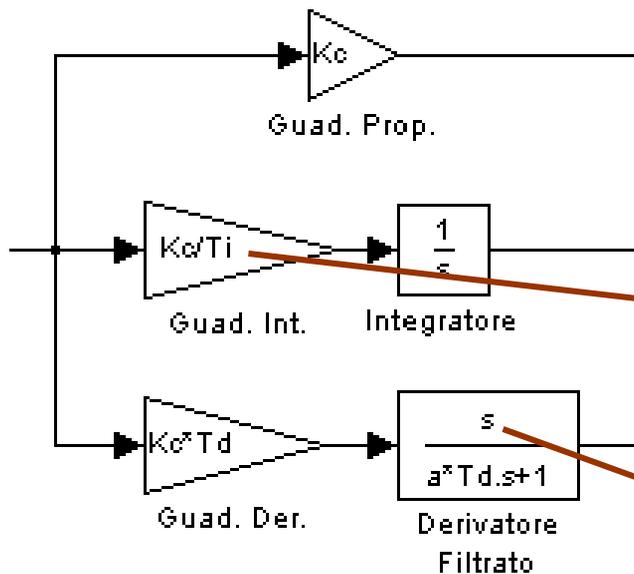
Initial step size:



Costruzione di un regolatore PID

- Costruire il modello di un regolatore PID

$$R_{PID}(s) = K_C \left(1 + \frac{1}{T_I s} + T_D s \right)$$



Parameters

Icon shape:

List of signs:

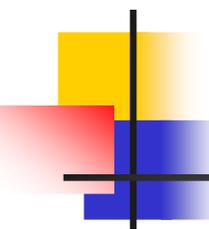
Parameters

Gain:

Parameters

Numerator:

Denominator:



Contenuti

- Introduzione a Matlab
- Presentazione Control System Toolbox
- Introduzione a Simulink
- **Esempi**
 - **Controllo di un doppio integratore**

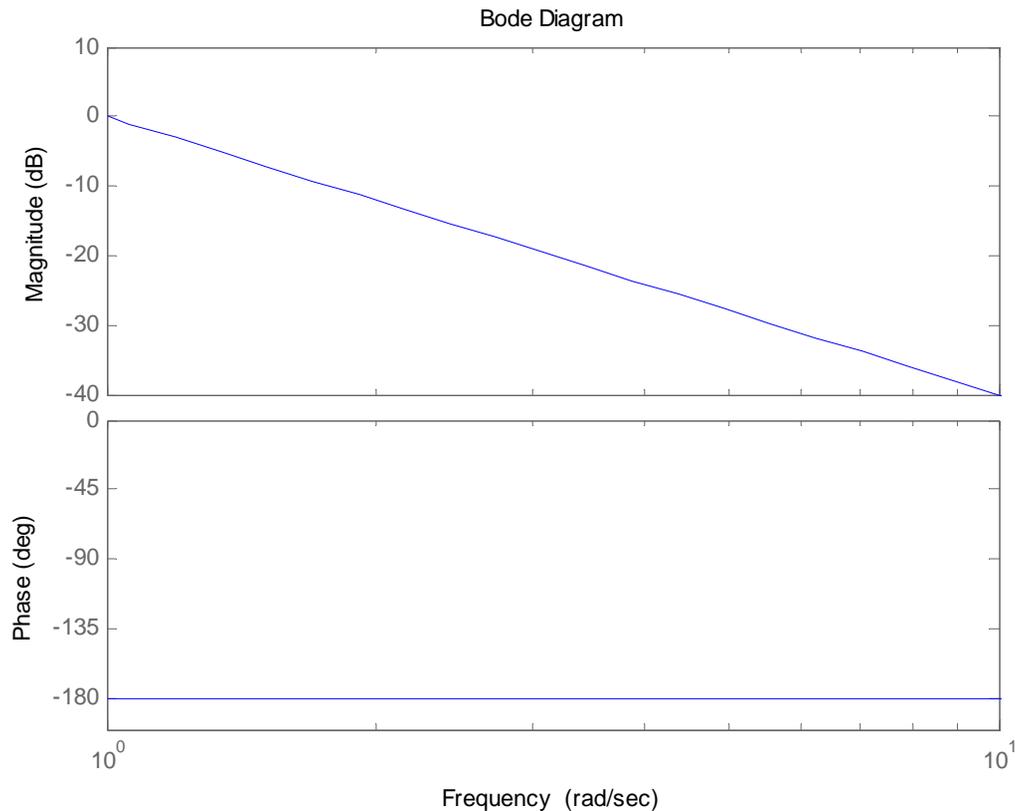
Controllo di un doppio integratore

- Modello del sistema da controllare

$$G(s) = \frac{1}{s^2}$$

» `G = tf(1,[1 0 0]);`

» `bode(G)`



Taratura del regolatore (1)

- Regolatore PID (scolastico)

$$R_{PID}(s) = K_C \left(1 + \frac{1}{T_I s} + T_D s \right) = \frac{K_C (T_I T_D s^2 + T_I s + 1)}{T_I s}$$

- Si sceglie $T_I = 4T_D$ per avere due zeri coincidenti

$$R_{PID}(s) = \frac{K_C \left(\frac{1}{4} T_I^2 s^2 + T_I s + 1 \right)}{T_I s} = \frac{K_C \left(\frac{1}{2} T_I s + 1 \right)^2}{T_I s}$$

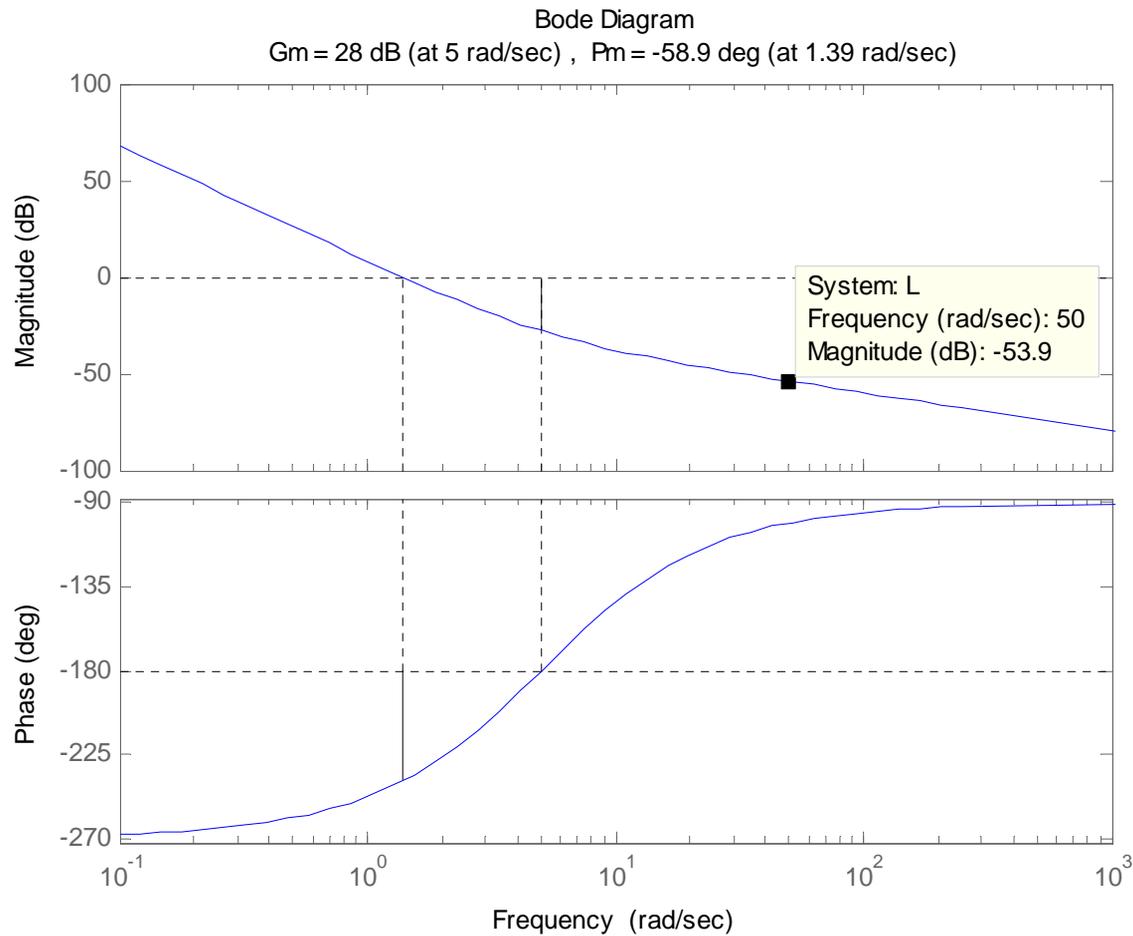
- Funzione d'anello

$$L(s) = R_{PID}(s) \cdot G(s) = \frac{K_C \left(\frac{1}{2} T_I s + 1 \right)^2}{s^3 T_I}$$

Taratura del regolatore (2)

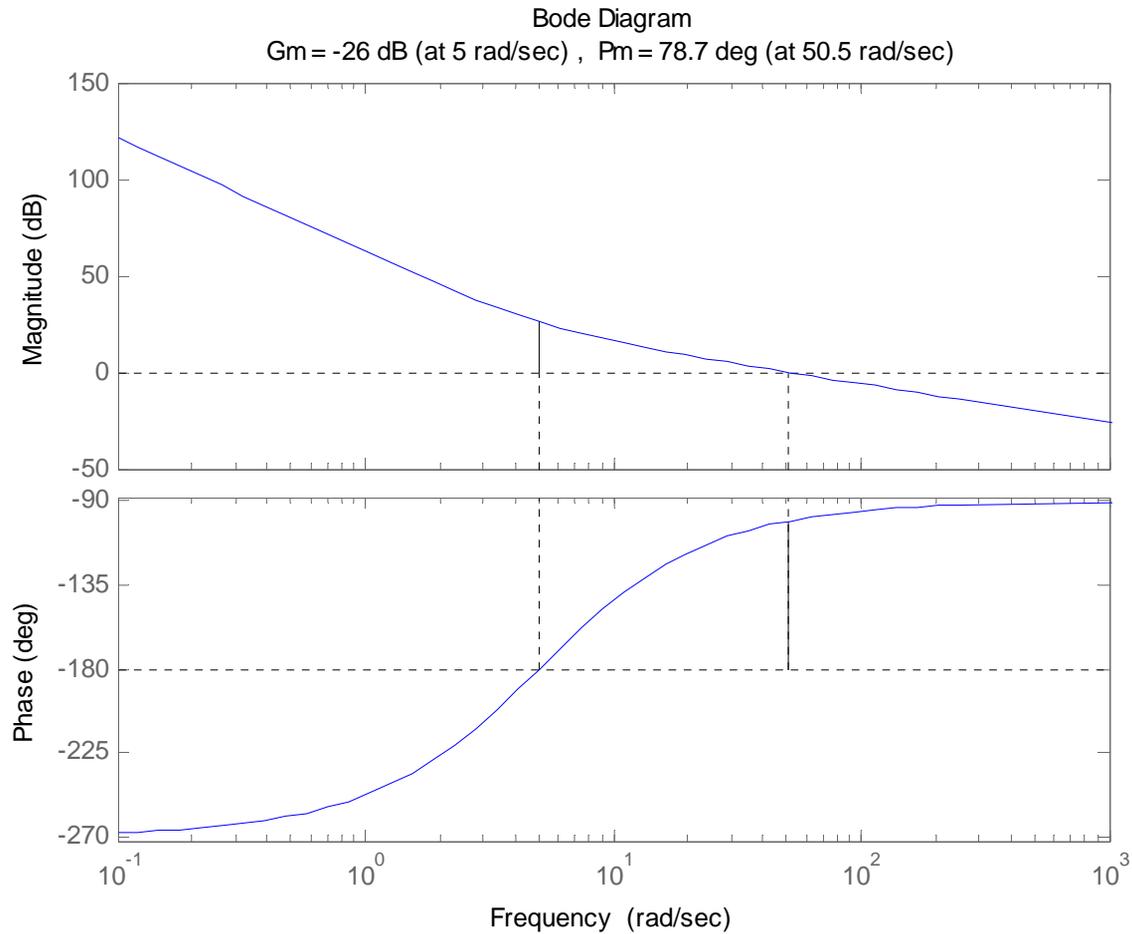
- Si sceglie T_I per posizionare gli zeri all'incirca una decade prima della banda passante richiesta (es $\omega_c=50 \text{ rad/s} \Rightarrow \omega_z=5 \text{ rad/s}$)
 $T_I = 0.4 \quad T_D = 0.1$
- Si "aggiusta" il guadagno per ottenere la banda passante desiderata
 - » $K_c = 1; T_i = 0.4; T_d = 0.1;$
 - » $R = \text{tf}(K_c*[T_i*T_d \ T_i \ 1],[T_i \ 0])$
 - » $L = R*G$ % Funzione d'anello
 - » $\text{margin}(L)$

Taratura del regolatore (3)

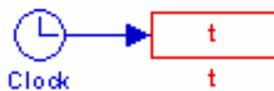
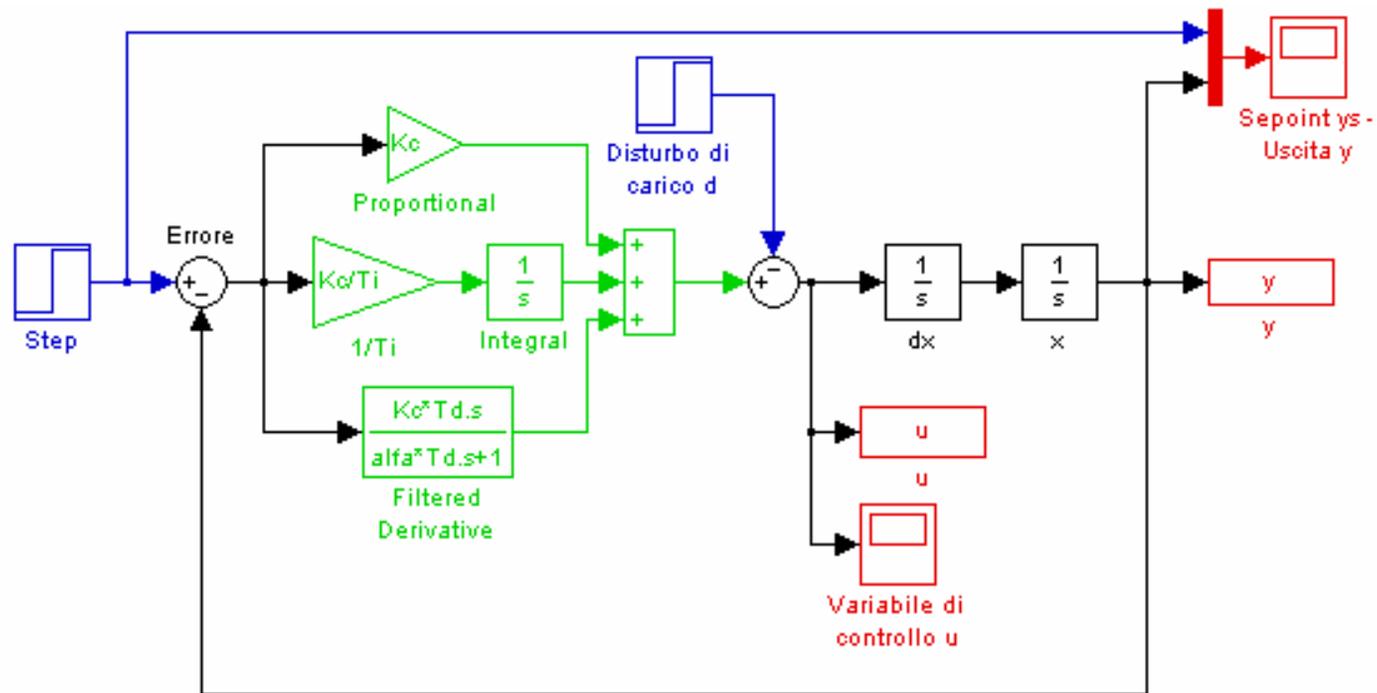


$$K_C = 10^{\frac{|L(j50)|}{20}} \cong 500$$

Taratura del regolatore (4)

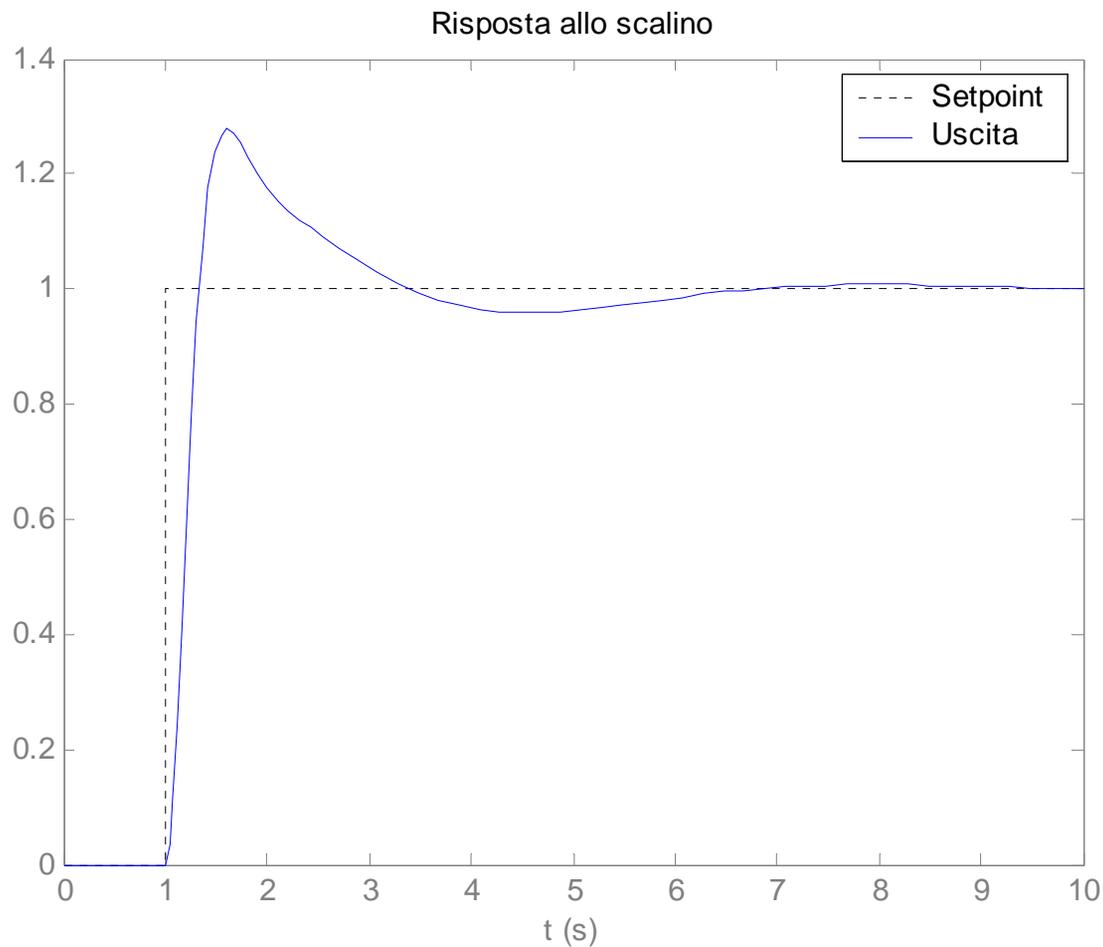


Sistema Simulink



Esempi di simulazioni (1)

■ Risposta allo scalino



Esempi di simulazioni (2)

- Risposta al disturbo di carico

