Utilizzo di Matlab per l'analisi di sistemi dinamici lineari

Sistemi dinamici lineari
a tempo continuo
ed a tempo discreto

Indice del materiale

- **Analisi** e **simulazione** di **sistemi dinamici LTI** in ambiente Matlab
  - Rappresentazione di sistemi dinamici lineari tempo-invarianti nell'ambiente Matlab
  - Analisi e simulazione di sistemi dinamici lineari tempo-invarianti

- **Sistemi lineari interconnessi**
  - Sistemi dinamici lineari interconnessi in ambiente Matlab
  - Utilizzo dell'istruzione Matlab FEEDBACK

Analisi e simulazione di sistemi dinamici LTI in ambiente Matlab

Definizioni
Proprietà

Sistemi dinamici lineari

- Un sistema dinamico **lineare tempo-invariante (LTI system)** può essere descritto:
  - in forma di **equazioni di stato**, assegnando le **matrici A,B,C,D**

$$\begin{cases} \dot{x} = Ax + Bu \\ y = Cx + Du \end{cases}$$

$$\begin{cases} x(k+1) = Ax(k) + Bu(k) \\ y(k) = Cx(k) + Du(k) \end{cases}$$

Sistemi dinamici lineari (...)

- Un sistema dinamico **lineare tempo-invariante (LTI system)** può essere descritto:
  - in forma di **matrice di funzioni di trasferimento**
    - mediante i **polinomi a numeratore e denominatore** delle funzioni di trasferimento

$$H(s) = \frac{b_m s^m + b_{m-1} s^{m-1} + \ldots b_0}{a_n s^n + a_{n-1} s^{n-1} + \ldots a_0}$$

$$H(z) = \frac{b_m z^m + b_{m-1} z^{m-1} + \ldots b_0}{a_n z^n + a_{n-1} z^{n-1} + \ldots a_0}$$

Sistemi dinamici lineari (...)

- Un sistema dinamico **lineare tempo-invariante (LTI system)** può essere descritto:
  - in forma di **matrice di funzioni di trasferimento**
    - assegnando **zeri, poli e guadagno** delle funzioni di trasferimento

$$H(s) = K \frac{\prod_q (s + z_q)}{\prod_r (s + p_r)}$$

$$H(z) = K \frac{\prod_q (z + z_q)}{\prod_r (z + p_r)}$$

Sistemi dinamici lineari (…)

- Nell'ambiente Matlab è possibile definire un sistema dinamico lineare tempo-invariante come ***oggetto*** **di tipo LTI model** a partire da qualsiasi di queste descrizioni.

Sistemi dinamici lineari (…)

- Utilizzando questi oggetti di tipo **LTI model** è possibile **analizzare le proprietà** del sistema dinamico corrispondente (stabilità ecc.) ed è possibile **simulare l'evoluzione nel tempo** del sistema dinamico, con condizioni iniziali ed ingressi assegnati.

Sistemi lineari in forma di equazioni di stato

- partendo dalle **equazioni di stato** (sia **SISO** che **MIMO**)

  – definire le matrici **A,B,C,D** nel **workspace**;
  – definire il sistema mediante il comando ***ss***

  – **sintassi** del comando **ss**
    - ***SYS = ss(A,B,C,D)*** crea un sistema dinamico a tempo continuo

    - ***SYS = ss(A,B,C,D,Ts)*** crea un sistema dinamico a tempo discreto, con intervallo di campionamento specificato da Ts [s].
    - Ponendo **Ts** pari a **-1**, si lascia non specificato l'intervallo di campionamento associato al sistema.

Esempi (1)

- Definizione del sistema dinamico a tempo continuo:

$$\begin{cases} \dot{x}_1 = -x_2 + 3u \\ \dot{x}_2 = -3x_1 + 2x_2 \\ y = 4x_1 + 2u \end{cases}$$

```
» A = [ 0 -1;-3 2]; B = [3;0];
    C = [4 0];  D = 2;

» sistema = ss(A,B,C,D)

a =
     x1  x2
 x1   0  -1
 x2  -3   2
b =
     u1
 x1   3
 x2   0
c =
     x1  x2
 y1   4   0
d =
     u1
 y1   2
Continuous-time model.
```

Esempi (2)

- Definizione del sistema dinamico a tempo discreto:

$$\begin{cases} x_1(k+1) = -x_2(k) + 3u(k) \\ x_2(k+1) = -3x_1(k) + 2x_2(k) \\ y(k) = 4x_1(k) + 2u(k) \end{cases}$$

```
» A = [ 0 -1; -3  2]; B = [3;0];
    C = [4 0];  D = 2;

» sistema = ss(A,B,C,D,-1)
a =
     x1  x2
  x1   0  -1
  x2  -3   2
b =
     u1
  x1   3
  x2   0
c =
     x1  x2
  y1   4   0
d =
     u1
  y1   2
Sampling time: unspecified
Discrete-time model.
```

Esempi (3)

- Definizione del sistema dinamico MIMO a tempo discreto:

$$\begin{cases} x_1(k+1) = -x_2(k) + 3u_1(k) - 2u_2(k) \\ x_2(k+1) = -3x_1(k) + 2x_2(k) - u_2(k) \\ y(k) = 4x_1(k) + 2u_1(k) \end{cases}$$

```
» A = [ 0 -1; -3  2]; B = [3 -2;0 -1];
    C = [4 0];  D = [2 0];

» sistema = ss(A,B,C,D,-1)
a =
     x1  x2
  x1  0  -1
  x2 -3   2
b =
     u1  u2
  x1  3  -2
  x2  0  -1
c =
     x1  x2
  y1  4   0
d =
     u1  u2
  y1  2   0

Sampling time: unspecified
Discrete-time model.
```

Sistemi lineari in forma di matrice di FdT

- caso **SISO**: partendo dalla **funzione di trasferimento**
  - assegnare i coefficienti dei polinomi a numeratore e denominatore della fdt nel workspace (nel seguito vettori **NUM** e **DEN**);
  - definire il sistema mediante il comando ***tf***
  - **Sintassi** del comando ***tf***
    - ***SYS = tf(NUM,DEN)*** crea un sistema a tempo continuo
    - ***SYS = tf(NUM,DEN,Ts)*** crea un sistema a tempo discreto con intervallo di campionamento specificato da Ts [s].
    - Ponendo **Ts** pari a **-1**, si lascia non specificato l'intervallo di campionamento associato al sistema.

Esempi (3)

- Definizione del sistema tramite la funzione di trasferimento

$$G(s) = \frac{s+1}{s^2+3s+16}$$

```
» num = [ 1 1 ]; den = [ 1 3 16 ];
» sistema = tf( num, den )
 Transfer function:
   s + 1
-------------
s^2 + 3 s + 16
```

Esempi (4)

- Definizione del sistema tramite la funzione di trasferimento

$$G(z) = \frac{z+1}{z^2+3z+16}$$

```
» num = [ 1 1 ]; den = [ 1 3 16 ];
» sistema = tf( num, den,-1 )
 Transfer function:
   z + 1
-------------
z^2 + 3 z + 16
```

Sistemi lineari in forma di matrice di FdT (2)

- Ancora caso **SISO**: partendo dalla **funzione di trasferimento**
  - assegnare i vettori degli zeri, dei poli ed il guadagno del sistema nel workspace (nel seguito vettori **Z**, **P** e **K**);
  - definire il sistema mediante il comando ***zpk***.
  - **Sintassi** del comando ***zpk***
    - ***SYS = zpk(Z,P,K)*** crea un sistema a tempo continuo
    - ***SYS = zpk(Z,P,K,Ts)*** crea un sistema a tempo discreto con intervallo di campionamento specificato da Ts [s].
    - Ponendo **Ts** pari a **-1**, si lascia non specificato l'intervallo di campionamento associato al sistema.

Esempi (5)

- Definizione del sistema tramite la funzione di trasferimento

$$G(s) = -5\frac{s-1}{s+1}$$

```
» Z = [ 1 ]; P = [ -1]; K = [-5]
» sistema = zpk( Z,P,K )
» Zero/pole/gain:
-5 (s-1)
--------
 (s+1)
```

Esempi (6)

- Definizione del sistema tramite la funzione di trasferimento

$$G(z) = -5\frac{z-1}{z+1}$$

```
» Z = [ 1]; P = [ -1]; K = [-5]
» sistema =zpk( Z, P, K ,-1)
 Zero/pole/gain:
-5 (z-1)
--------
 (z+1)
 Sampling time: unspecified
```

Sistemi lineari in forma di matrice di FdT (3)

- Comando ***tf*** nel caso **MIMO**: partendo dalla **matrice di funzioni di trasferimento**, per sistemi con ***NU*** ingressi ed ***NY*** uscite
  - i vettori ***NUM*** e ***DEN*** diventano in realtà *cell array* di dimensione NY*NU
  - gli elementi ***NUM{i, j}*** e ***DEN{i, j}*** specificano in tal caso la FdT dall'ingresso j-esimo verso l'uscita i-esima.
  - esempio:

H = tf({-5 ; [1 -5 6]} , {[1 -1] ; [1 1 0]})

crea la matrice di FdT per il sistema a 2 uscite ed 1 ingresso descritto da

$$H(s) = \begin{bmatrix} -\dfrac{5}{s-1} \\ \dfrac{s^2-5s+6}{s^2+s} \end{bmatrix}$$

Sistemi lineari in forma di matrice di FdT (4)

- Comando ***zpk*** nel caso **MIMO**: partendo dalla **matrice di funzioni di trasferimento**, per sistemi con ***NU*** ingressi ed ***NY*** uscite
  - i vettori ***Z*** e ***P*** diventano cell array di dimensione NY * NU
  - gli elementi ***Z{i, j}*** e ***P{i, j}*** specificano zeri e poli della FdT dall'ingresso j-esimo all'uscita i-esima;
  - K è la matrice delle costanti di guadagno: K(i, j) è la costante di guadagno per la FdT tra l'ingresso j-esimo e l'uscita i-esima;
  - Esempio:

H = zpk({[];[2 3]} , {1;[0 -1]} , [-5;1])

descrive il sistema

$$H(s) = \begin{bmatrix} -\dfrac{5}{s-1} \\ \dfrac{(s-2)(s-3)}{s(s+1)} \end{bmatrix}$$

Simulazione in Matlab di sistemi lineari

- Funzioni disponibili per la simulazione:
  - ***impulse*** ⇒ simulazione risposta all'impulso;
  - ***step*** ⇒ simulazione risposta a scalino;
  - ***initial*** ⇒ simulazione movimento libero;
  - ***lsim*** ⇒ simulazione con ingresso qualsiasi e stato iniziale qualsiasi.

Simulazione in Matlab di sistemi lineari (2)

- Sintassi:

```
» [y,t]=step(sistema);
» [y,t]=step(sistema,t);
```

Vettore d'uscita

Vettore dei tempi

```
» [y,x]=lsim(sistema,u,t);
```

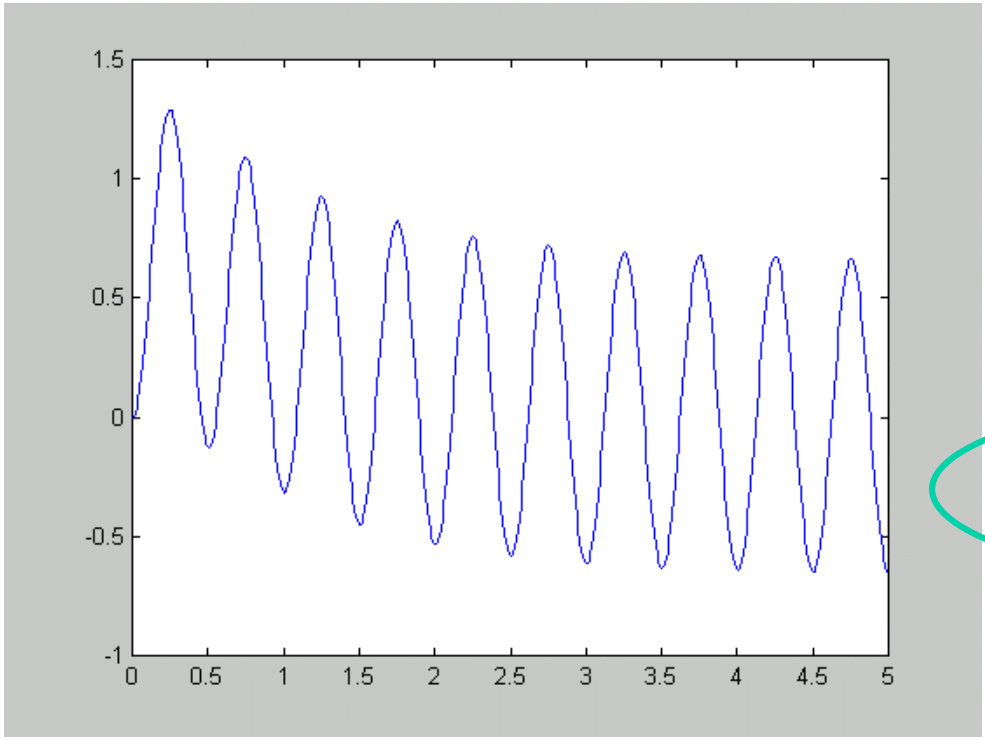
Vettore sequenza d'ingresso

Vettore d'uscita

Vettore di stato

Vettore dei tempi

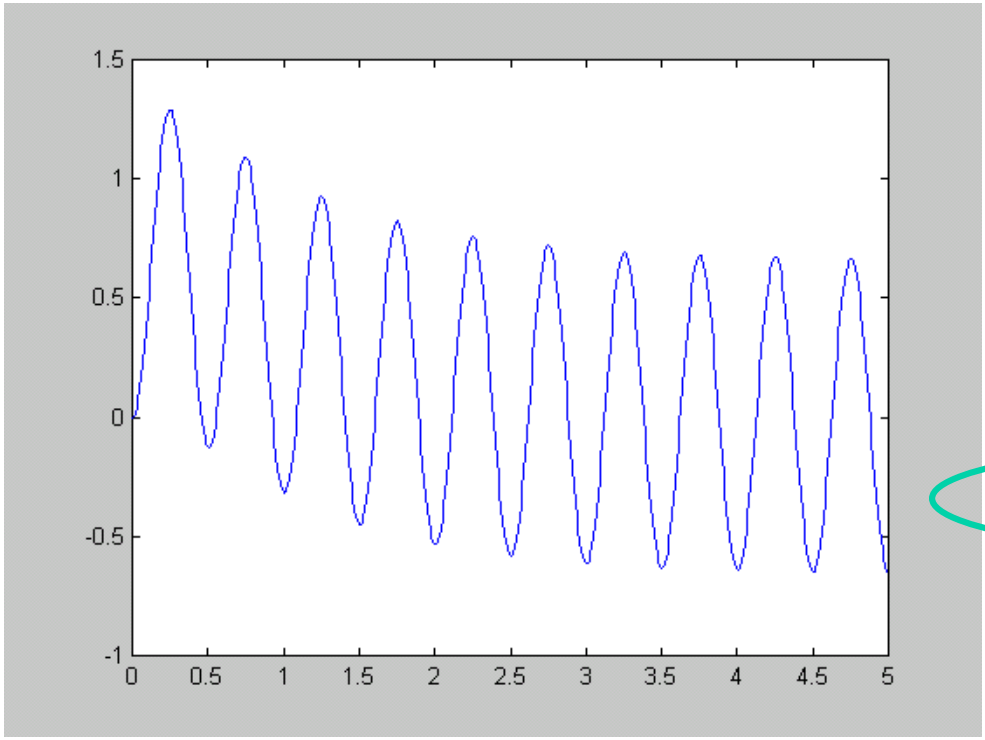
Esempio di utilizzo (1)




```
» a=[-1 ,0;3,-4];
» b=[2;1];c=[1,2];d=0;
» sistema=ss(a,b,c,d);
» t=(0:0.01:5);
» u=2*sin(2*pi*2*t);
» y=lsim(sistema,u,t);
» plot(t,y);
```

Il risultato della simulazione è stato assegnato ad una variabile e successivamente visualizzato in un grafico.

Esempio di utilizzo (2)




```
» a=[-1 ,0;3,-4];
» b=[2;1];c=[1,2];d=0;
» sistema=ss(a,b,c,d);
» t=(0:0.01:5);
» u=2*sin(2*pi*2*t);
» lsim(sistema,u,t);
```

Utilizzando le funzioni senza assegnare il risultato della simulazione a variabili d'uscita si ottiene direttamente il grafico dell'evoluzione temporale.

Esempi di utilizzo (3)

- Analisi della risposta allo scalino unitario:

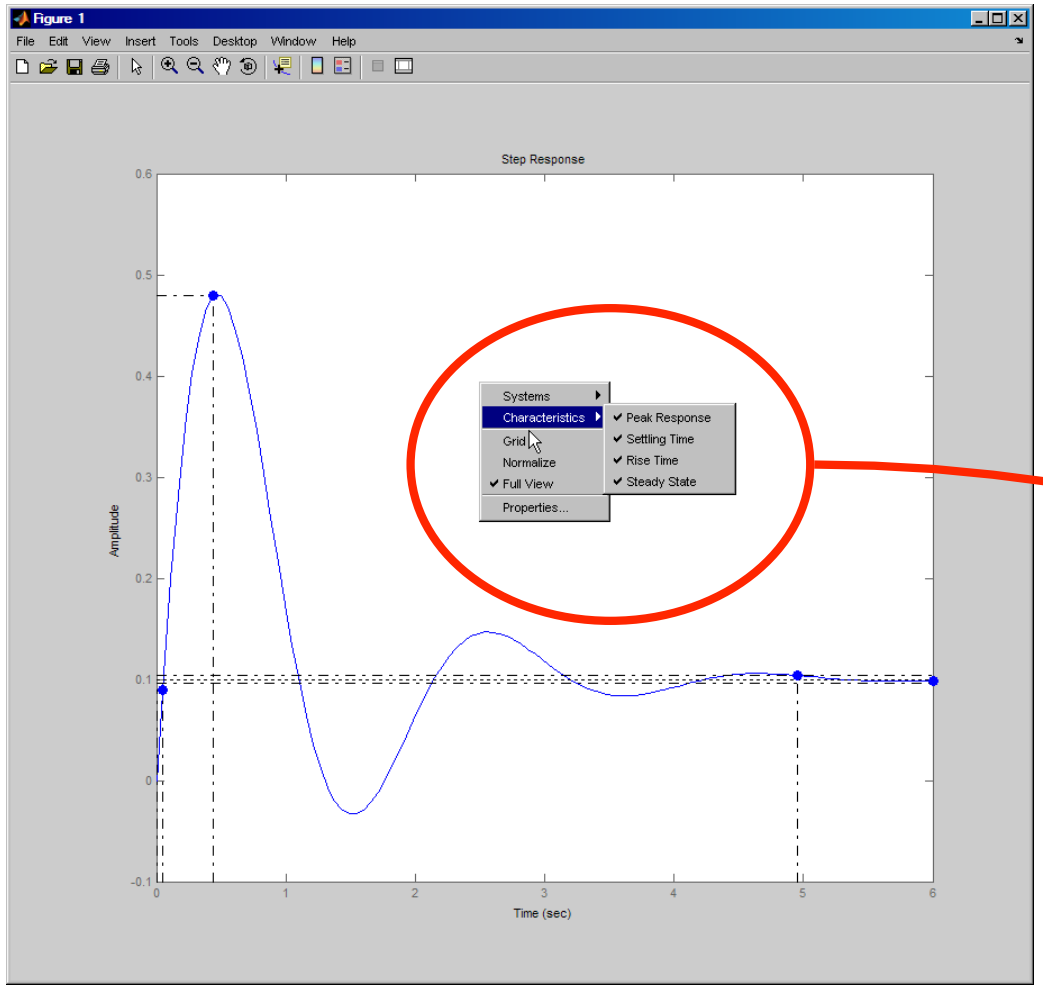
» step(sistema);

- Interagendo tramite il mouse nella finestra che visualizza l'andamento della risposta allo scalino è possibile ottenere informazioni relative a
  - Sovraelongazione della risposta
  - Valore di regime
  - Tempo di salita (*rise time*)
  - Tempo di assestamento (*settling time*)

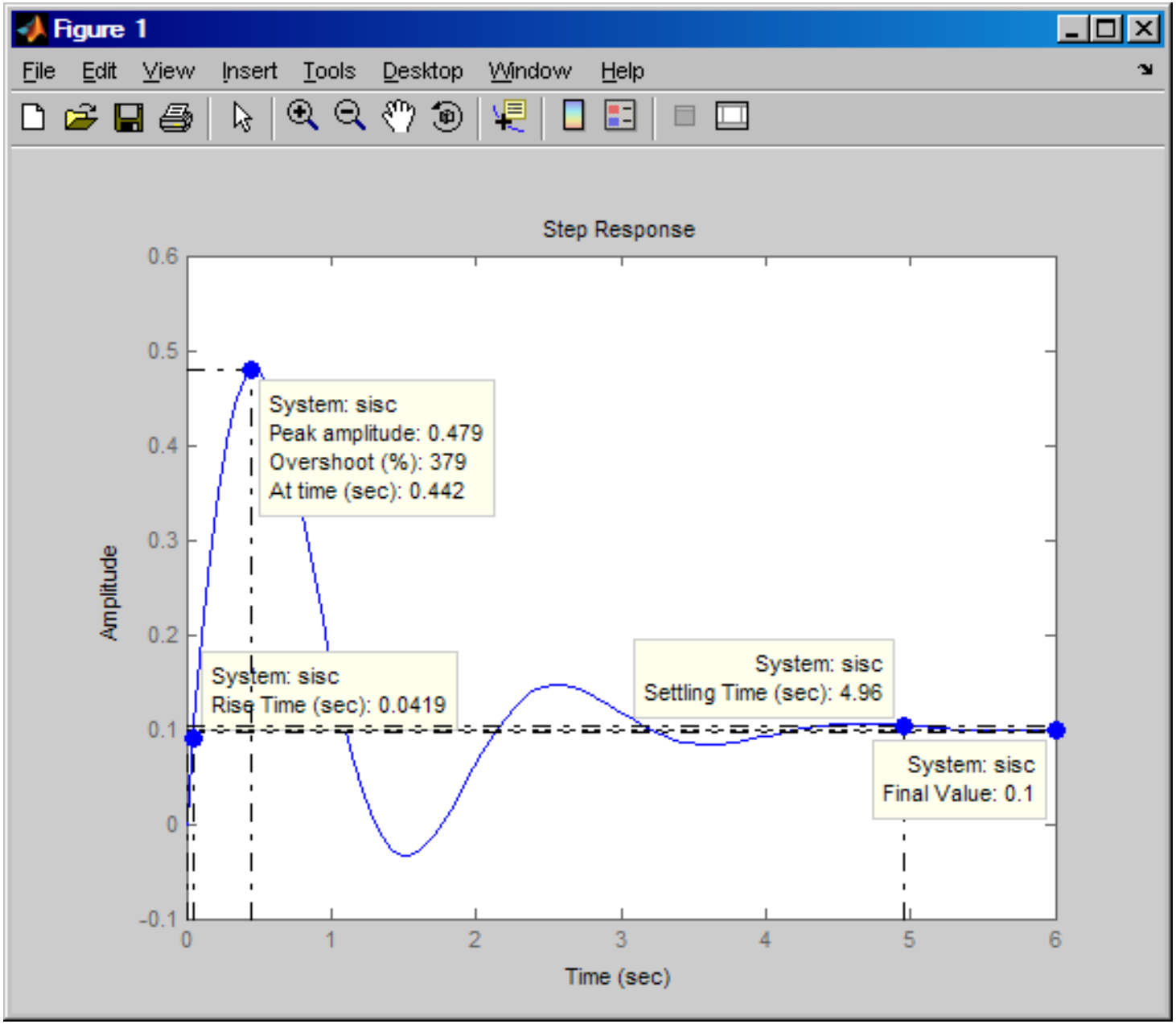
Esempio: analisi della risposta allo scalino

- Il codice Matlab dell'esempio:

```
» % sistema a tempo continuo
» sisc = tf([2 1],[1 2 10]);
»
» % sistema a tempo discreto
» sisd = tf([2 1],[1 0.2 0.5], 0.1);
»
» figure; step(sisc);
» figure;step(sisd);
```

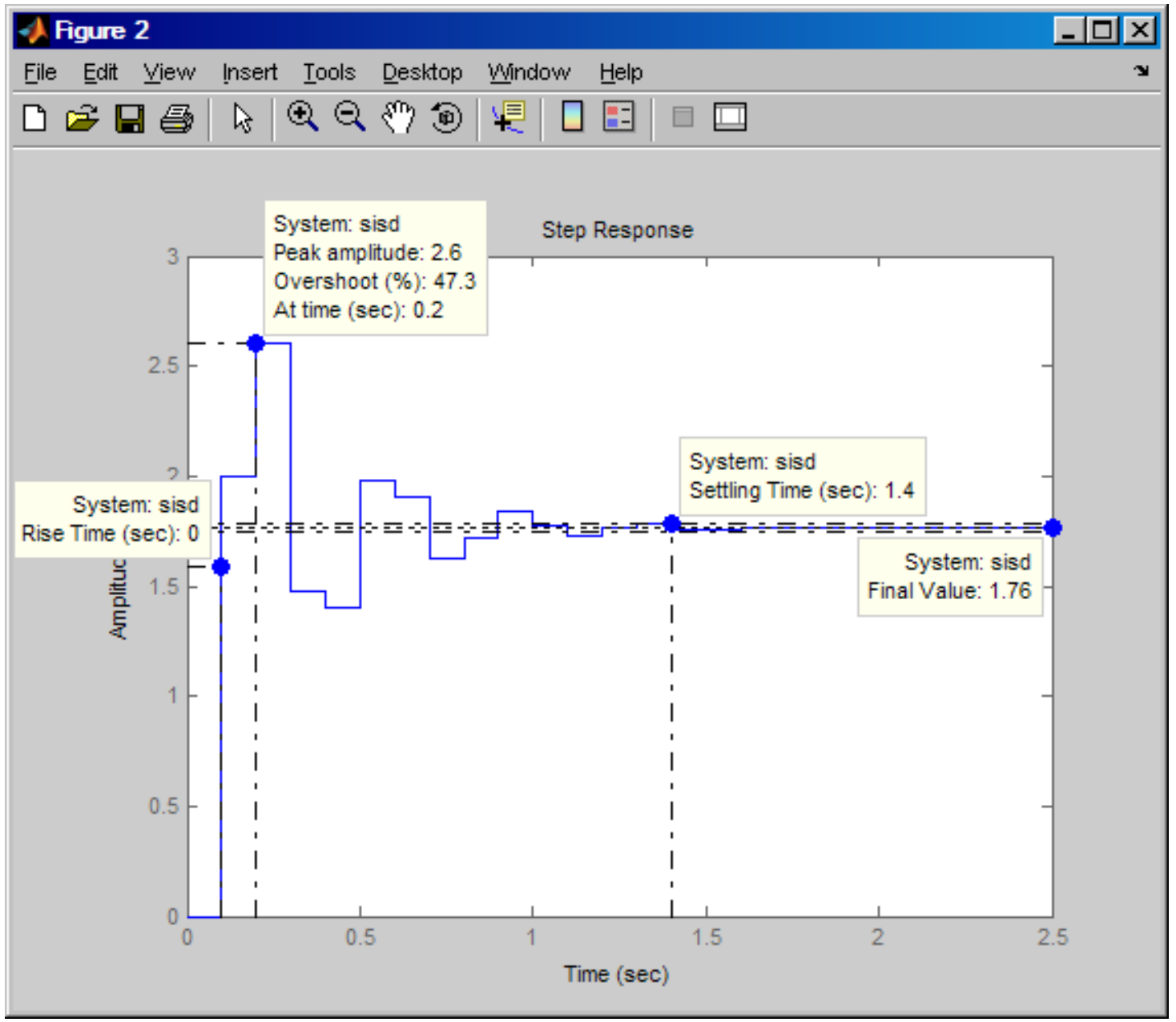


A seguito di un "click" del pulsante destro del mouse compare un menu …



Esempio: risposta del sistema a tempo continuo.

Esempio: risposta del sistema a tempo discreto.



Considerazioni riassuntive

- Come tutti i risultati di **operazioni di calcolo numerico**, anche i sistemi LTI, descritti con le istruzioni Matlab viste, possono essere affetti da **errori** (errata o mancante cancellazione di termini nella FdT ecc. …).

- Si vedano l'argomento "**Reliable Computations**" e l'argomento "**Choice of LTI Model**" nella documentazione del **Control Toolbox**.

Sistemi lineari interconnessi

Definizioni, proprietà, applicazioni

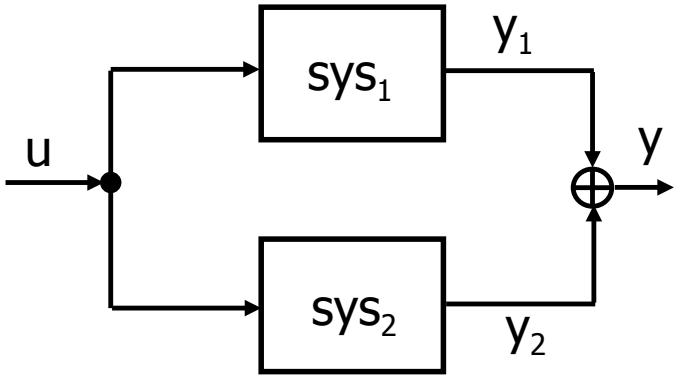
Interconnessione di sistemi

- Agli *oggetti* sistemi lineari si possono applicare i normali operatori **+**,*,**/**, **** con il seguente significato:
  - **+** connessione in **parallelo**;
  - ***** connessione in **serie**;
  - **/, ** usati per definire l'operazione di inversione (a sx, a dx) per **sistemi quadrati**.

Esempi di interconnessioni elementari

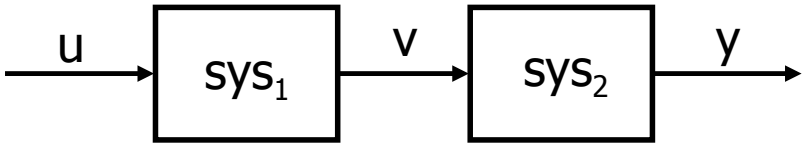
- **Connessione parallelo**

$$sys = sys_1 + sys_2$$



- **Connessione serie**

$$sys = sys_1 \cdot sys_2$$



Altri esempi di operazioni elementari

- **Inversione** (per sistemi quadrati)

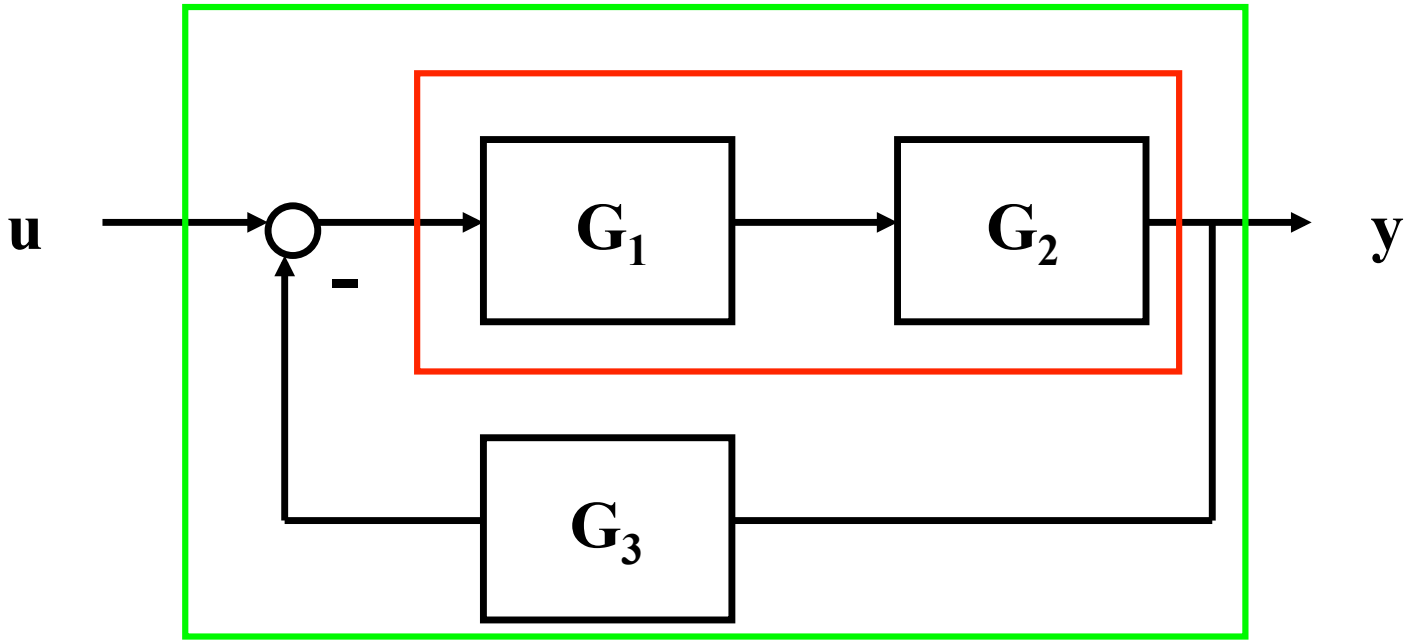
$$sys_1 \setminus sys_2 \quad \Leftrightarrow \quad inv(sys_1) \cdot sys_2$$

$$sys_1 / sys_2 \quad \Leftrightarrow \quad sys_1 \cdot inv(sys_2)$$

- **Trasposizione**

$$sys^T \Leftrightarrow sys.'$$

Esempio di connessione



andata = g1*g2; retroazione = andata/(1+andata*g3)

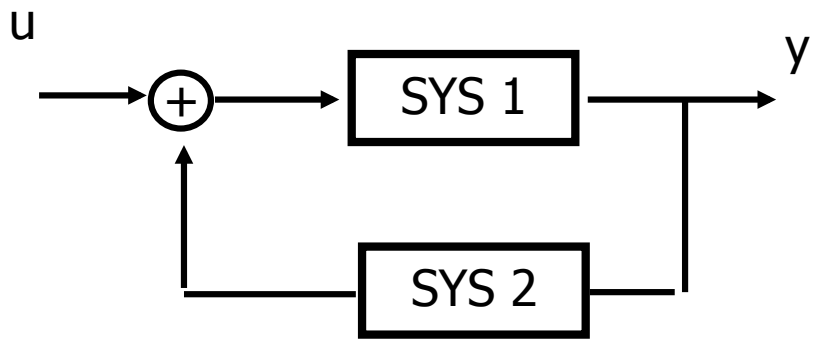
Interconnessioni tra sistemi

- Funzioni che permettono di descrivere interconnessioni tra sistemi dinamici:
  - concatenazione **,** (orizz.) **;** (vert.)
  - connessione diagonale a blocchi **append**
  - connessione parallelo di due blocchi **parallel**
  - connessione serie di due blocchi **series**
  - connessione in retroazione **feedback**
  - connessione generica **connect**

Sintassi del comando feedback

Sintassi semplice del comando

SYS = FEEDBACK(SYS1,SYS2) fornisce il sistema LTI corrispondente al semplice ciclo di reazione seguente

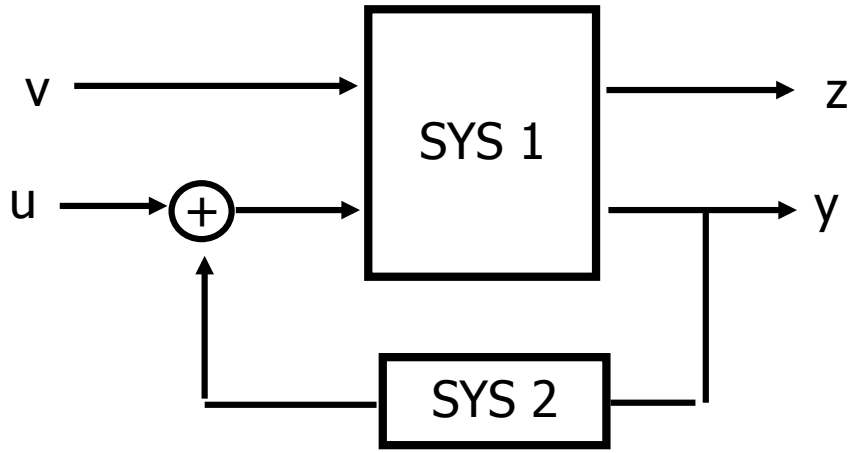


Si presuppone che sia uno schema a **retroazione negativa**. Nel caso in cui si voglia una retroazione positiva, va specificato nel comando

SYS = FEEDBACK(SYS1,SYS2,+1).

Sintassi completa

SYS = FEEDBACK(SYS1,SYS2,FEEDIN,FEEDOUT,SIGN) fornisce il sistema LTI corrispondente alla struttura:



I vettori **FEEDIN** e **FEEDOUT** contengono rispettivamente gli indici degli ingressi e delle uscite del sistema SYS1 coinvolti nella retroazione. La variabile **SIGN** indica se si tratta di retroazione positiva (**SIGN** pari a +1) oppure negativa (**SIGN** pari a –1 oppure non assegnato).