



Sommario

- Modalità di rappresentazioni di sistemi dinamici lineari.
- Funzioni per le rappresentazioni tramite funzione di trasferimento, matrici A,B,C,D e zeri-poli-guadagno.

Si ringrazia Alessandro Di Giorgio per le slide della presente lezione



Metodi di rappresentazione di un sist. dinamico lineare (1/2)

- Siamo interessati a sistemi:
 - Lineari.
 - A tempo continuo o a tempo discreto.
 - Stazionari (o tempo-invarianti).



Metodi di rappresentazione di un sist. dinamico lineare (2/2)

- Un sistema dinamico lineare stazionario a dimensione finita può essere descritto:
 - In forma di funzione di trasferimento, attraverso la definizione dei due polinomi $N(s)$ e $D(s)$.
 - In forma di variabili di stato mediante le quattro matrici A, B, C, D .
 - In forma zeri-poli-guadagno.
- In MATLAB possiamo definire i sistemi lineari come oggetti a partire dalle tre rappresentazioni mostrate sopra.



Rappresentazione tramite f. di trasf. (1/2)

- Data la funzione di trasferimento:

$$G(s) = \frac{N(s)}{D(s)}$$

- Possiamo costruire un oggetto che descrive il sistema attraverso la funzione `tf`.
- `sys = tf(num, den)` crea una funzione di trasferimento a tempo continuo con numeratore e denominatore specificato dai parametri `num` e `den`.



Rappresentazione tramite f. di trasf. (2/2)

Esempio:

$$G(s) = \frac{s + 2}{s^2 + 2s + 5}$$

```
num = [1 2]; den = [1 2 5];
```

```
sys = tf(num,den)
```

Transfer function:

$s + 2$

$s^2 + 2s + 5$



Rappresentazione con matrici A,B,C,D (1/2)

- Data la rappresentazione del sistema tramite matrici A,B,C,D seguente:

$$\dot{x} = Ax + Bu$$

$$y = Cx + Du$$

- Possiamo costruire un oggetto che descrive il sistema attraverso la funzione `ss`.
- `sys = ss(A,B,C,D)` crea un oggetto che rappresenta un modello nello spazio di stato di un sistema a tempo continuo. In input vengono passate le 4 matrici A,B,C,D.



Rappresentazione con matrici A,B,C,D (2/2)

Dato il sistema seguente:

$$\dot{x} = -x + 2u$$

$$y = 3x + u$$

$$A = -1;$$

$$B = 2;$$

$$C = 3;$$

$$D = 1;$$

$$\text{sys} = \text{ss}(A, B, C, D)$$

Output di
MATLAB

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



Rappresentazione Zero-Pole-Gain (1/2)

- Data la rappresentazione del sistema in forma zeri-poli-guadagno seguente:

$$G(s) = K \frac{(s - z_1)(s - z_2) \dots (s - z_n)}{(s - p_1)(s - p_2) \dots (s - p_n)}$$

- Possiamo costruire un oggetto che descrive il sistema attraverso la funzione `zpk`.
- `sys = zpk(z, p, k)` crea un oggetto che rappresenta un sistema a tempo continuo passando in input gli zeri `z`, i poli `p` e il guadagno `k`.



Rappresentazione Zero-Pole-Gain (2/2)

Esempio:

$$G(s) = 2 \frac{s - 5}{(s - 4 + i)(s - 4 - i)}$$

```
zeros = [5]; poles = [4-i 4+i]; gain = 2;  
sys = zpk(zeros,poles,gain)
```

Zero/pole/gain:

2 (s-5)

(s^2 - 8s + 17)



Sistemi a tempo discreto (1/3)

- E' possibile utilizzare le funzioni mostrate finora anche per modellare sistemi dinamici a tempo discreto.
- E' sufficiente aggiungere, come ultimo parametro, il tempo di campionamento.
- Se come tempo di campionamento viene passato -1 il tempo di campionamento viene considerato non specificato.



Sistemi a tempo discreto (2/3)

- Ad esempio, se vogliamo imporre 0.1 come tempo di campionamento e costruire un sistema a partire dalle matrici A,B,C,D, scriviamo, in MATLAB:

```
A_d=[1 1; 2 1];  
B_d=[1; 6];  
C_d=[1 3];  
D_d=[1];  
sys_sd=ss(A_d, B_d, C_d, D_d, 0.1)
```

Output di
MATLAB

```
a =  
      x1  x2  
x1      1   1  
x2      2   1  
  
b =  
      u1  
x1      1  
x2      6  
  
c =  
      x1  x2  
y1      1   3  
  
d =  
      u1  
y1      1  
  
Sampling time: 0.1  
Discrete-time model.
```



Sistemi a tempo discreto (3/3)

- Infine, con il comando `c2d` è possibile convertire un sistema a tempo continuo in uno a tempo discreto.
- E' necessario passare come parametri (i) il sistema a tempo continuo; (ii) il tempo di campionamento; (iii) il metodo con il quale si vuole effettuare la conversione. E' possibile scegliere tra varie alternative, ad esempio:
 - `'zoh'`: Zero-order hold. Molto usato e garantisce la corrispondenza tra i campioni di ingresso e di uscita tra il sistema a tempo continuo e a tempo discreto negli istanti di campionamento.
 - `'tustin'`: Bilinear (Tustin) approximation. Tipicamente usato per convertire leggi di controllo ricavate nel continuo.
- Un esempio di invocazione è il seguente:

```
sys_discreto=c2d(sys_continuo,0.1,'zoh').
```



Sessione di studio



Esercitazione

Provare a rappresentare in MATLAB le seguenti funzioni di trasferimento:

- $G(s) = \frac{1}{s+1}$
- $G(s) = \frac{1-s+s^2}{s^3+3s+2}$
- $G(s) = \frac{s}{s+2}$
- $G(s) = \frac{s(s-1)}{s^4+1}$



Sessione di studio



Esercitazione

Provare a rappresentare in MATLAB le seguenti rappresentazioni nello spazio di stato:

- $\dot{x} = -2x + u; y = \frac{1}{2}x$
- $A = \begin{bmatrix} 1 & 2 \\ 2 & 1 \end{bmatrix}, B = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, C = [1 \quad 1], D = 1$



Sessione di studio



Esercitazione

Provare a rappresentare in MATLAB i seguenti sistemi dinamici lineari definiti attraverso zeri, poli e guadagno:

- $G(s) = 2.5 \frac{(s-1)(s+1)}{(s-1+i)(s-1-i)(s+3)}$
- $G(s) = \frac{s}{(s+1)(s+2)}$