Introduzione a MATLAB

Principali comandi **MATLAB** utili per il corso di

Fondamenti di Automatica 01AYS

Politecnico di Torino

Sistemi dinamici LTI

1. Simulazione a tempo continuo

Definizione del sistema

- Per creare sistemi dinamici lineari tempo invarianti LTI, a tempo continuo, si possono seguire due strade:

  - un modello ingresso-stato-uscita può essere definito a partire dalla forma in variabili di stato, cioè dalle matrici A, B, C, D;
  - un modello ingresso-uscita può essere definito a partire dalla funzione di trasferimento del sistema.

Modello ingresso-stato-uscita

- Partendo dalla rappresentazione in variabili di stato

$$\begin{cases} \dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{u} \\ \boldsymbol{y} = \boldsymbol{C}\boldsymbol{x} + \boldsymbol{D}\boldsymbol{u} \end{cases}$$

  - si definiscono le matrici A, B, C, D;
  - a partire dalle matrici A, B, C, D si definisce l'oggetto sistema con il comando:

```
>> SYS = ss(A,B,C,D)
```

  - ✓ Si può definire D=0 per creare una matrice nulla della dimensione appropriata.

Modello ingresso-uscita (1)

- Definizione "polinomiale" della funzione di trasferimento:
  - si definiscono i polinomi che rappresentano il numeratore ed il denominatore;
  - si definisce la funzione di trasferimento con il comando tf(num,den).
- tf crea la funzione di trasferimento a partire dai vettori num e den:

```
>> SYS = tf(NUM,DEN)
```

  - ✓ num, den sono i vettori contenenti i coefficienti delle potenze di s in ordine decrescente:

num=[3 2 1] equivale a: num= 3s^2+2s+1

Modello ingresso-uscita (2)

- Definizione "simbolica" della funzione di trasferimento:
  - si definisce direttamente la variabile s:

```
>> s=tf('s')
>> fdt=(1+s)/(3*s^2+5*s-2)
```

Legami fra le rappresentazioni (1)

- Per calcolare la funzione di trasferimento di un sistema a partire dalla sua rappresentazione in variabili di stato si utilizza il comando **SS2TF** :

```
>>[NUM,DEN] = ss2tf(A,B,C,D,k)
```

  - ✓ I vettori num e den contengono i coefficienti delle potenze decrescenti di s.
  - ✓ Per i sistemi MIMO è possibile calcolare la k-esima colonna della matrice di trasferimento.
  - ✓ Per i sistemi SISO, non occorre specificare k.

Legami fra le rappresentazioni (2)

- Per calcolare la funzione di trasferimento di un oggetto sistema SYS di tipo SISO si utilizza il comando **tfdata**:

```
>>[NUM,DEN] = tfdata(SYS,'v')
```

  - ✓ I vettori riga num e den contengono i coefficienti delle potenze di s in ordine decrescente.

Simulazione (1)

- **LSIM** simula la risposta di un sistema LTI ad un ingresso arbitrario.

  Per tracciare la risposta del sistema SYS all'ingresso definito da U (matrice degli ingressi) e T (vettore del tempo) si utilizza:

```
>> lsim(SYS,U,T)
```

  ove U ha tante colonne quanti sono gli ingressi e la k-esima riga di U contiene i campioni degli ingressi all'istante T(k).

  Ad esempio:

```
>> T = 0:0.01:5;
>> U = sin(T);
>> lsim(SYS,U,T)
```

Simulazione (2)

- Per simulare sistemi con stato iniziale assegnato X0:

```
>> lsim(SYS,U,T,X0)
```

- Per riportare l'andamento della risposta di più sistemi LTI su un unico grafico:

```
>> lsim(SYS1,SYS2,...,U,T,X0)
```

Simulazione (3)

- Per memorizzare l'evoluzione dell'uscita YS ed il vettore dei tempi TS:

```
>>[YS,TS] = lsim(SYS,U,T)
```

- Per memorizzare l'evoluzione degli stati XS, dell'uscita YS ed il vettore dei tempi TS:

```
>>[YS,TS,XS] = lsim(SYS,U,T)
```

  - ✓ In questi ultimi due casi non viene tracciato alcun grafico. Naturalmente i grafici desiderati possono essere visualizzati con il comando plot, come nell'esempio seguente.

Simulazione (4): esempio

- Esempio

```
>>SYS=ss(A,B,C,D);
>>[YS,TS,XS]=lsim(SYS,U,T,X0);
>>figure(1),plot(TS,XS)
>>figure(2),plot(TS,YS)
```

Simulazione (5)

Risposta di sistemi LTI ad ingressi canonici:

- risposta al gradino:
  - ✓ **step(SYS)** simula la risposta al gradino con scelta automatica del numero di istanti temporali e della durata della simulazione;
  - ✓ **step(SYS,TFINAL)** simula la risposta al gradino da t=0 a t=TFINAL, con scelta automatica degli istanti temporali intermedi;
  - ✓ **step(SYS,T)** simula la risposta al gradino rispetto al vettore del tempo T precedentemente definito;
  - ✓ **step(SYS1,SYS2,T)** riporta l'andamento della risposta al gradino di più sistemi su un singolo grafico rispetto al vettore del tempo T precedentemente definito;

Simulazione (6)

- **[Y,T] = step(SYS)** memorizza l'evoluzione dell'uscita Y ed il vettore dei tempi T.

  In questo caso non viene tracciato alcun grafico.

- **[Y,T,X] = step(SYS)** memorizza l'evoluzione dell'uscita Y, degli stati X e del vettore dei tempi T.

  In questo caso non viene tracciato alcun grafico.

  Naturalmente i grafici desiderati possono essere visualizzati con il comando plot.

Simulazione (7)

- risposta all'impulso:
  - **impulse(SYS)** simula la risposta all'impulso con scelta automatica del numero di istanti temporali e della durata della simulazione;
  - **impulse(SYS,TFINAL)** simula la risposta all'impulso da t=0 a t=TFINAL, con scelta automatica degli istanti temporali;
  - **impulse(SYS,T)** simula la risposta all'impulso rispetto al vettore del tempo T precedentemente definito;
  - **impulse(SYS1,SYS2,T)** riporta l'andamento della risposta all'impulso di più sistemi su un singolo grafico rispetto al vettore del tempo T precedentemente definito;

Simulazione (8)

- **[Y,T] = impulse(SYS)** memorizza l'evoluzione dell'uscita Y ed il vettore dei tempi T.

  In questo caso non viene tracciato alcun grafico.

- **[Y,T,X] = impulse(SYS)** memorizza l'evoluzione dell'uscita Y, degli stati X e del vettore dei tempi T.

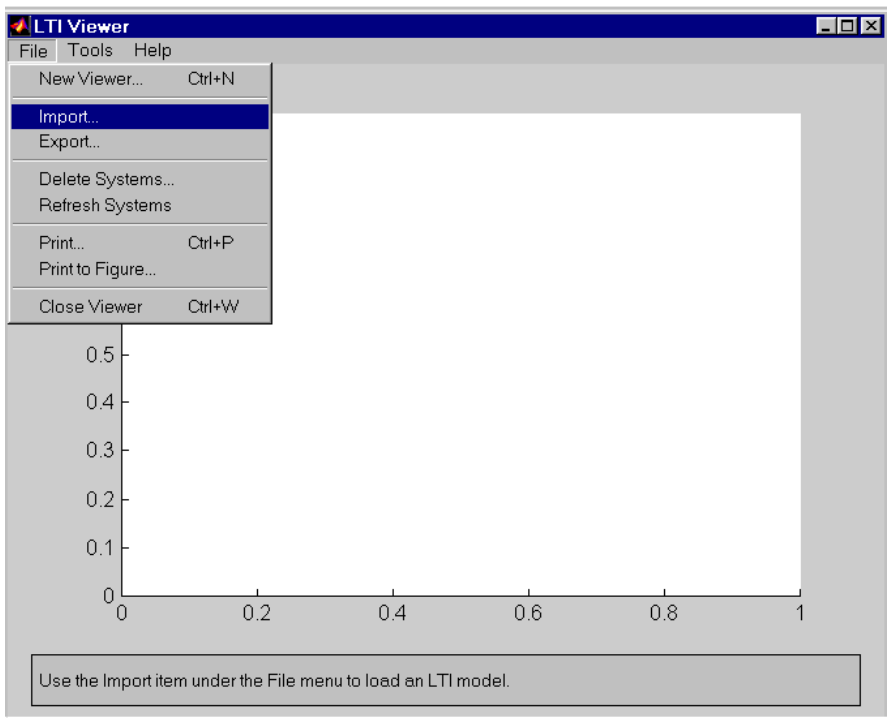
  In questo caso non viene tracciato alcun grafico.

  Naturalmente i grafici desiderati possono essere visualizzati con il comando plot.

LTI viewer (1)

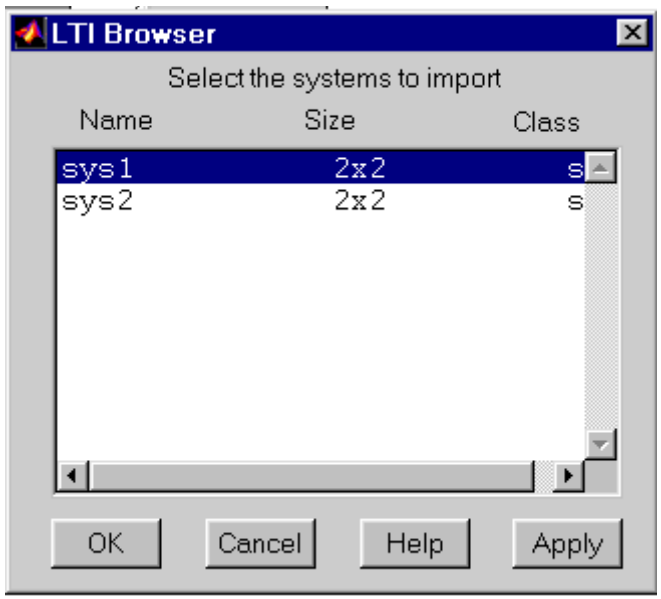
- **ltiview** apre una finestra per la simulazione di sistemi LTI. Per visualizzare la risposta dei sistemi definiti in precedenza, si procede come mostrato nella figura seguente:

```
>> ltiview
```



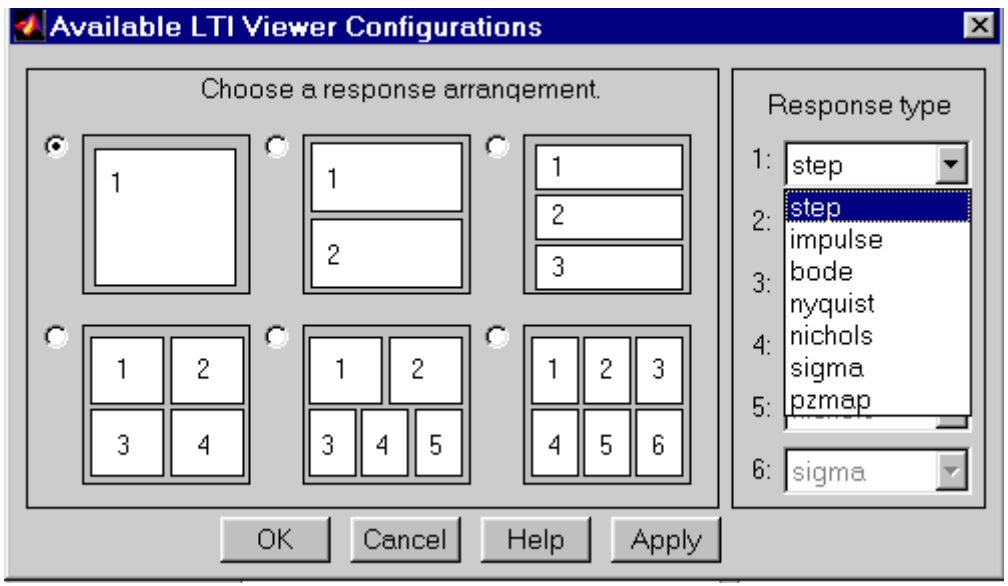
LTI viewer (2)

- In questo modo si può selezionare il sistema desiderato (i sistemi devono essere definiti in precedenza) .



LTI viewer (3)

- Selezionando Tools -> Viewer Configuration si può modificare l'ingresso applicato per la simulazione:



Sistemi dinamici LTI

2. Analisi di proprietà strutturali

Operazioni su matrici (1)

- **rank(A)**: calcola il rango della matrice A (numero di righe o colonne linearmente indipendenti).

  rank(A,tol) : numero dei valori singolari di A maggiori di tol.

```
>>   K=rank(A)
```

Operazioni su matrici (2)

- L'operatore di divisione fra matrici è \
  - A\B =(A)^-1*B .
  - X = A\B calcola la soluzione dell'equazione A*X = B.
  - Per ulteriori informazioni sul comando, basta digitare:

```
>> help mldivide
```

Raggiungibilità

- Per calcolare la matrice di raggiungibilità si utilizza il comando **ctrb**:
  - `R = ctrb(A,B)` calcola la matrice di raggiungibilità definita come [B AB A^2B ...].
  - `R = ctrb(SYS)` restituisce la matrice di raggiungibilità del sistema SYS precedentemente definito con il metodo delle variabili di stato.

Osservabilità

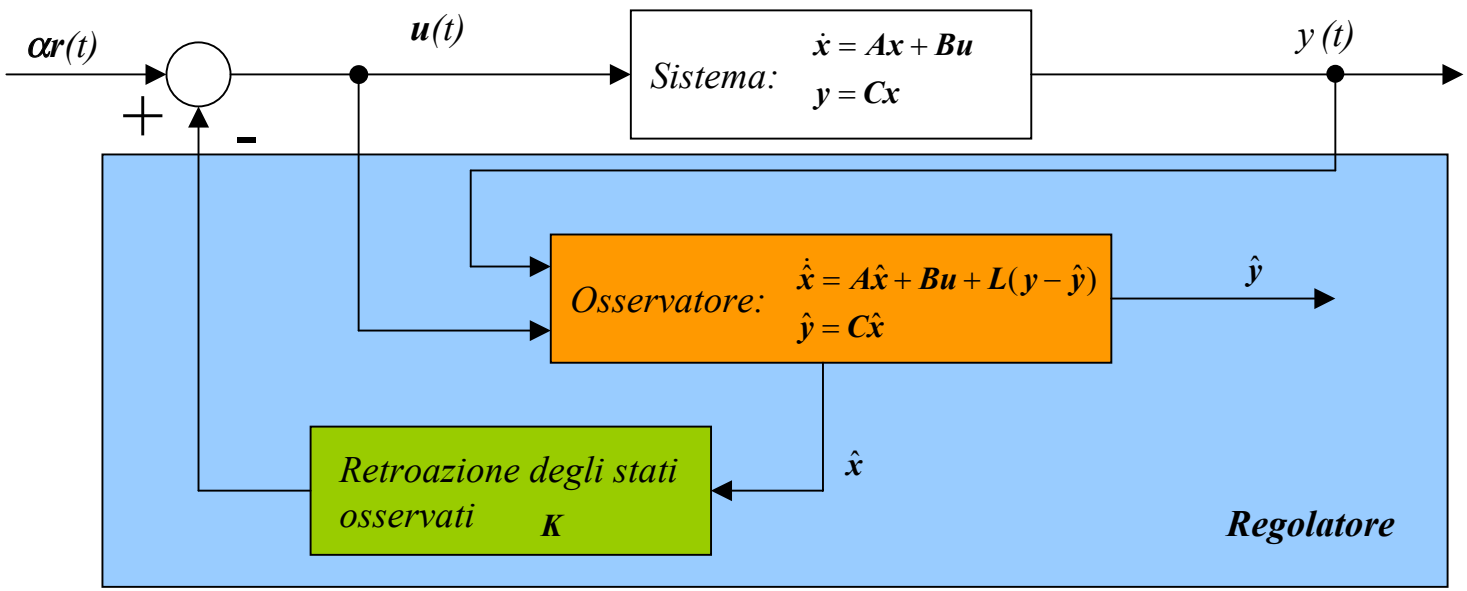
- Per calcolare la matrice di osservabilità si utilizza il comando **obsv**:
  - `O = obsv(A,C)` calcola la matrice di osservabilità definita come [C; CA; CA^2...].
  - `O = obsv(SYS)` restituisce la matrice di osservabilità del sistema SYS precedentemente definito con il metodo delle variabili di stato.

Sistemi dinamici LTI

3. Progetto del regolatore per sistemi SISO

Struttura del regolatore

- Per progettare il regolatore è necessario posizionare opportunamente gli autovalori λ_i di (A-BK) ed (A-LC).



Posizionamento dei λ_i (A-BK)

- Se il sistema è completamente raggiungibile, è possibile assegnare arbitrariamente tutti gli autovalori di (A-BK) definendo un vettore P contenente gli autovalori desiderati e utilizzando i comandi **place** ed **acker**:

```
>> K = place(A,B,P)
```

oppure

```
>> K = acker(A,B,P)
```

  - NB: non è possibile assegnare autovalori coincidenti con il comando place.

Posizionamento dei λ_i (A-LC)

- Se il sistema è completamente osservabile, è possibile assegnare arbitrariamente tutti gli autovalori di (A-LC) per dualità, definendo un vettore P contenente gli autovalori desiderati e utilizzando i comandi **place** ed **acker**:

```
>> L = place(A',C',P)'
```

oppure

```
>> L = acker(A',C',P)'
```

  - NB: non è possibile assegnare autovalori coincidenti con il comando place.

Posizionamento degli autovalori

- Al termine, si consiglia di verificare sempre il corretto posizionamento degli autovalori con il comando:

```
>> P1 = eig(A-BK)
>> P2 = eig(A-LC)
```