

## Esercizi introduttivi a Matlab

Inserire le seguenti matrici

$$A = \begin{bmatrix} 1 & 2 & 3 \\ -4 & -3 & 2 \\ 0 & 0 & 3 \end{bmatrix} \quad b = \begin{bmatrix} 2 \\ 0 \\ 1 \end{bmatrix} \quad c = \begin{bmatrix} 0 & 1 & 1 \end{bmatrix}$$

```
A=[1 2 3;-4 -3 2;0 0 3]
b=[1; 0; 1] oppure b=[1 0 1]'
c=[0 1 1]
```

Nota: per non “vedere” il risultato mettere ; al termine dell’istruzione.

Calcolare  $A*b$ .

```
A*b
```

Valutare la matrice trasposta di  $A$ .

```
A'
```

Estrarre l’elemento in posizione (2,3) di  $A$ .

```
A(2,3)
```

Estrarre dalla matrice  $A$  la seconda colonna e la terza riga.

```
A(:,2)
```

```
A(3,:)
```

Calcolare gli autovalori della matrice  $A$  e il suo polinomio caratteristico.

```
autovalori=eig(A)
```

```
coeff=poly(A)
```

Pertanto  $\Delta_A(\lambda) = \lambda^3 - \lambda^2 - \lambda - 15$

Le radici del polinomio caratteristico  $\Delta_A(\lambda)$  sono gli autovalori: `roots(coeff)`

Valutare il determinante della matrice  $A$  e la sua matrice inversa.

```
det(A)
```

```
inv(A)
```

Valutare la dimensione di  $b$ .

```
[r,c]=size(b)
```

Scrivere la matrice identità e nulla di dimensione pari alla dimensione di  $A$ .

```
eye(size(A))
```

```
zeros(size(A))
```

Scrivere un vettore di dimensione (10,1) i cui elementi sono tutti pari a 3.

```
3*ones(10,1)
```

Estrarre il primo e l’ultimo elemento del vettore  $b$ .

```
b(1)
```

```
b(length(b)) oppure b(end)
```

Creare un vettore i cui elementi variano da 0 a 100 con passo 0.01 oppure composto da 50 valori.  
[0:0.01:100]  
linspace(0,100,50)

Disegnare su uno stesso grafico le funzioni

$$y_1 = x + \frac{7}{1+x^2}$$

$$y_2 = x^2 - 2$$

con  $x \in [-5, 3]$ .

```
x=[-5:0.01:3];
```

```
y1=-x+7./(1+x.^2);
```

```
y2=x.^2-2;
```

```
figure; hold on; grid;
```

```
plot(x,y1);
```

```
plot(x,y2);
```

## Esercizi introduttivi al primo laboratorio

Sia dato il sistema lineare a tempo continuo

$$\dot{x}_1 = -2x_1 + 3x_2 - u$$

$$\dot{x}_2 = -x_1 + x_2 + 3u$$

$$y = x_1 - x_2$$

Scrivere le matrici  $(A, b, c, d)$  che caratterizzano il sistema.

```
A=[-2 3;-1 1];
```

```
b=[-1 3]';
```

```
c=[1 -1];
```

```
d=0;
```

Verificare l'esistenza e unicità dell'equilibrio corrispondente a ingresso costante  $\bar{u}$ .

`eig(A)` oppure `det(A)`

Determinare lo stato di equilibrio  $\bar{x}$  e l'uscita di equilibrio  $\bar{y}$  corrispondente a  $\bar{u} = 5$ .

```
ubar=5
```

```
xbar=inv(A)*b*ubar
```

```
ybar=c*xbar+d*ubar
```

Verificare la asintotica stabilità del sistema. Determinare il tempo di risposta.

```
real(eig(A))
```

```
reallD=max(real(eig(A)))
```

```
TD=-1/reallD
```

```
TR=5*TD
```

Definire il sistema "in variabili di stato".

```
sistema=ss(A,b,c,d)
```

Nota: per sistema a tempo discreto si ha `sistema=ss(A,b,c,d,1)`

Determinare la funzione di trasferimento  $G(s)$  del sistema, valutandone zeri, poli e guadagno.

```
[num,den]=ss2tf(A,b,c,d)
```

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

```
zero(G) oppure roots(num)
```

```
pole(G) oppure roots(den)
```

```
dcgain(G) oppure dcgain(sistema)
```

Simulare l'andamento dell'uscita del sistema e della seconda variabile di stato, a partire da condizione iniziale  $(1, -1)$  con ingresso costante nel tempo pari a 5.

```
tempo=linspace(0,2*TR,200);
```

```
u=5*ones(size(tempo));
```

```
x0=[1 -1]';
```

```
[y,t,x]=lsim(sistema,u,tempo,x0);
```

```
figure; plot(t,y);
```

```
figure; plot(t,x(:,2));
```

Simulare l'andamento dell'uscita del sistema con ingresso  $u(t) = 4\sin(\frac{2\pi}{3}t)$  e condizione iniziale nulla.

```
u=4*sin(2*pi*tempo/3);  
[y,t,x]=lsim(sistema,u,tempo);  
figure; plot(t,y);
```

Simulare la risposta a scalino (unitario).

step(sistema) oppure step(G) oppure step(A,b,c,d)

Simulare la risposta a impulso (unitario).

impulse(sistema) oppure impulse(G) oppure impulse(A,b,c,d)

Nota: l'impulso (o delta di Dirac) rappresenta matematicamente un segnale molto intenso che perdura per un tempo molto breve, con area sottesa unitaria. Per es, una forza impulsiva rappresenta un colpo che diamo alla pallina di Newton per imprimergli, da ferma, una velocità iniziale di 1 m/s.

Simulare la risposta del sistema corrispondente a condizione iniziale (1, -1) (ingresso nullo).

initial(sistema,x0)

Simulare la risposta del sistema corrispondente a ingresso costante pari a 5 e condizione iniziale pari a (1, -1) in modo alternativo rispetto a quanto già fatto, sfruttando il comando step e il principio di sovrapposizione.

```
y1=step(sistema,tempo);  
y2=initial(sistema,x0,tempo);  
figure; plot(tempo,5*y1+y2);
```

Nota: verificare, mediante un grafico, che il risultato ottenuto corrisponda a quello ottenuto in precedenza col comando lsim.

Mediante Simulink, simulare l'uscita del seguente sistema aggregato quando l'ingresso è pari a 5 e  $\mu$  è pari a +2 o -2.

Digitare simulink al prompt di Matlab e creare un nuovo modello con Blank Model.

Per rappresentare lo schema utilizzare il blocco Transfer Fcn per due volte (Library Browser → Simulink → Continuous → trascinare l'oggetto Transfer Fcn nel modello. Cliccare su blocco per inserire numeratore e denominatore della funzione di trasferimento).

Per sommare o sottrarre due segnali utilizzare il nodo sommatore (Commonly Used Blocks → Sum); per cambiare i segni del nodo sommatore occorre entrare nelle proprietà dello stesso e specificarli nel campo List of signs (da porre a |+-).

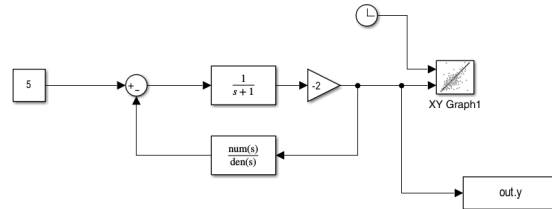
Per moltiplicare un segnale (guadagno  $\mu$ ) utilizzare il blocco Gain (Commonly Used Blocks → Gain. Cliccare sul blocco per cambiarne il valore).

L'ingresso, come richiesto dal testo, sarà una costante con valore 5 (Sources → Constant. Cliccare su blocco per cambiarne il valore). Porre sample time pari a 0.1 per simulare in tanti istanti di tempo.

Inserire il blocco Clock (Sources → Clock) per generare il tempo da inserire nel blocco XY (Sinks → XYGraph) in cui verrà visualizzata l'uscita del sistema.

Per ottenere l'uscita del sistema nel workspace di Matlab, collegare il blocco To Workspace (libreria sinks) all'uscita del sistema selezionando array alla voce Save format.

Collegare i blocchi inseriti secondo lo schema riportato nel testo dell'esercizio ottenendo la seguente figura:



Simulare il sistema per 15 unità di tempo (Stop Time nella finestra Simulation).

Per cambiare la visualizzazione del grafico (ottenibile con doppio click sul blocco XYGraph), selezionare la finestra Format del grafico.

Per estrarre il valore di regime dell'uscita nel caso  $\mu = 2$ , estrarre dalla variabile out la variabile y e valutarne il valore finale:

```
uscita=out.y;  
uscita(length(uscita)) oppure uscita(end)
```

## Esercizi introduttivi al secondo laboratorio

Verificare che il sistema è completamente raggiungibile.

```
R=ctrb(A,b)
det(R)
rank(R)
```

Determinare una legge di controllo  $K$  che assegni al sistema controllato gli autovalori in  $[-10 -20]$ .

```
autoK=[-10 -20];
K=acker(A,b,autoK)
eig(A-b*K)
```

Verificare che il sistema è completamente osservabile.

```
O=obsv(A,c)
det(O)
rank(O')
```

Determinare un ricostruttore asintotico dello stato che annulli a regime l'errore di ricostruzione in al più 5 unità di tempo.

```
autoL=[-1 -1];
L=acker(A',c',autoL)
eig(A'-c'*L)
```

## Esercizi introduttivi al terzo laboratorio

Tracciare il diagramma di Bode del sistema dato.

```
bode(G) oppure bode(sistema)
```

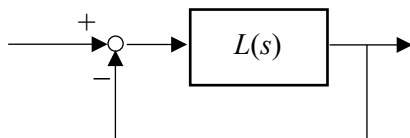
Determinare l'uscita del sistema a regime corrispondente a ingresso sinusoidale  $u(t) = 10\sin(2t)$ .

```
[mag,fase]=bode(sistema,2)
y=10*mag*sin(2*tempo+fase*2*pi/360);
figure; plot(tempo,y);
```

volendo sovrapporre la risposta da condizione iniziale nulla:

```
hold on;
u=10*sin(2*tempo);
[y,t,x]=lsim(sistema,u,tempo);
plot(t,y,'r');
```

Studiare la stabilità del sistema retroazionato



dove 
$$L(s) = \frac{10}{(1+10s)(1+s)(1+0.01s)}$$

```
num=10;
den=conv([10 1],conv([1 1],[0.01 1]));
L=tf(num,den);
margin(L)
```

## Comandi Matlab

help comando → sintassi sull'uso del comando

det → determinante di una matrice quadrata

eig → autovalori di una matrice quadrata

poly → coefficienti del polinomio caratteristico di una matrice quadrata elencati per potenza decrescente

roots → radici di un polinomio

inv → matrice inversa di una matrice quadrata

size → dimensione di una matrice

eye → matrice identità

zeros → matrice nulla

ones → matrice i cui elementi sono tutti pari a 1

length → lunghezza di un vettore

linspace → vettore di elementi equispaziati tra loro

figure → apre una nuova figura

plot → diagramma in una figura 2D i dati riportati in ascissa e ordinata

real → parte reale di un numero

isreal → controlla se un numero è reale (1) o complesso (0)

max → valore massimo

ss → definizione di un modello interno ((A,b,c,d) a tempo continuo, (A,b,c,d,1) a tempo discreto)

ss2tf → porta il sistema da modello interno a modello esterno (numeratore e denominatore della funzione di trasferimento)

tf → genera la funzione di trasferimento

zero → zeri della funzione di trasferimento

pole → poli della funzione di trasferimento

dcgain → guadagno della funzione di trasferimento

lsim → simula il sistema lineare

step → risposta a scalino da condizione iniziale nulla

impulse → risposta a impulso da condizione iniziale nulla

initial → risposta a ingresso nullo e condizione iniziale fissata

ctrb → matrice di raggiungibilità

obsv → matrice di osservabilità

rank → rango di una matrice

acker → comando per il posizionamento di autovalori in anello chiuso (con legge di controllo  $u = -kx$ )

bode → diagramma di bode

conv → prodotto di polinomi

margin → diagramma di Bode per lo studio della stabilità sei sistemi di controllo secondo criterio di Bode