

Fondamenti di Automatica

Introduzione a Matlab

Indice del materiale

- Descrizione generale di Matlab
- Alcune funzioni predefinite
- Definizione di matrici e vettori
- Definizione di polinomi
- Rappresentazione grafica dei dati
- Rappresentazione e simulazione di sistemi dinamici lineari

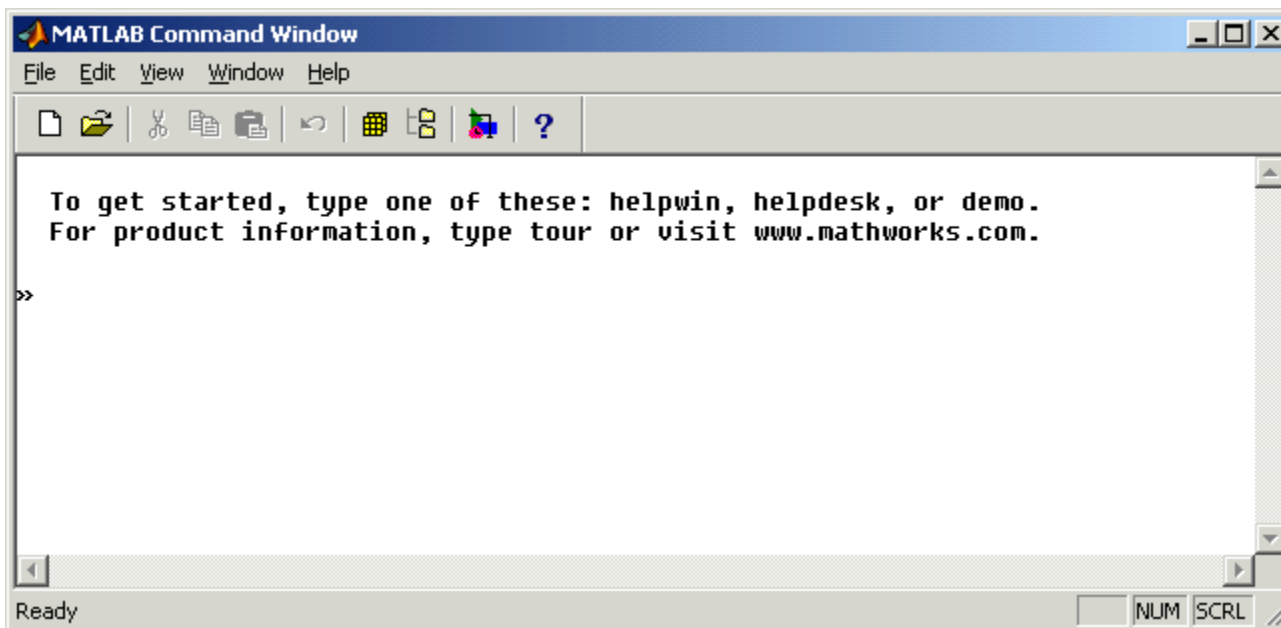
Descrizione generale di Matlab

MATLAB (= MATrix LABoratory)

- un linguaggio di programmazione per applicazioni scientifiche e numeriche
- vasto insieme di funzioni predefinite
- interprete di comandi
- possibilità di definire nuove funzioni
- libreria di TOOLBOX per svariate applicazioni (ad es. analisi dei sistemi e dei segnali, analisi e sintesi di sistemi di controllo, ecc.)

L'interfaccia di Matlab

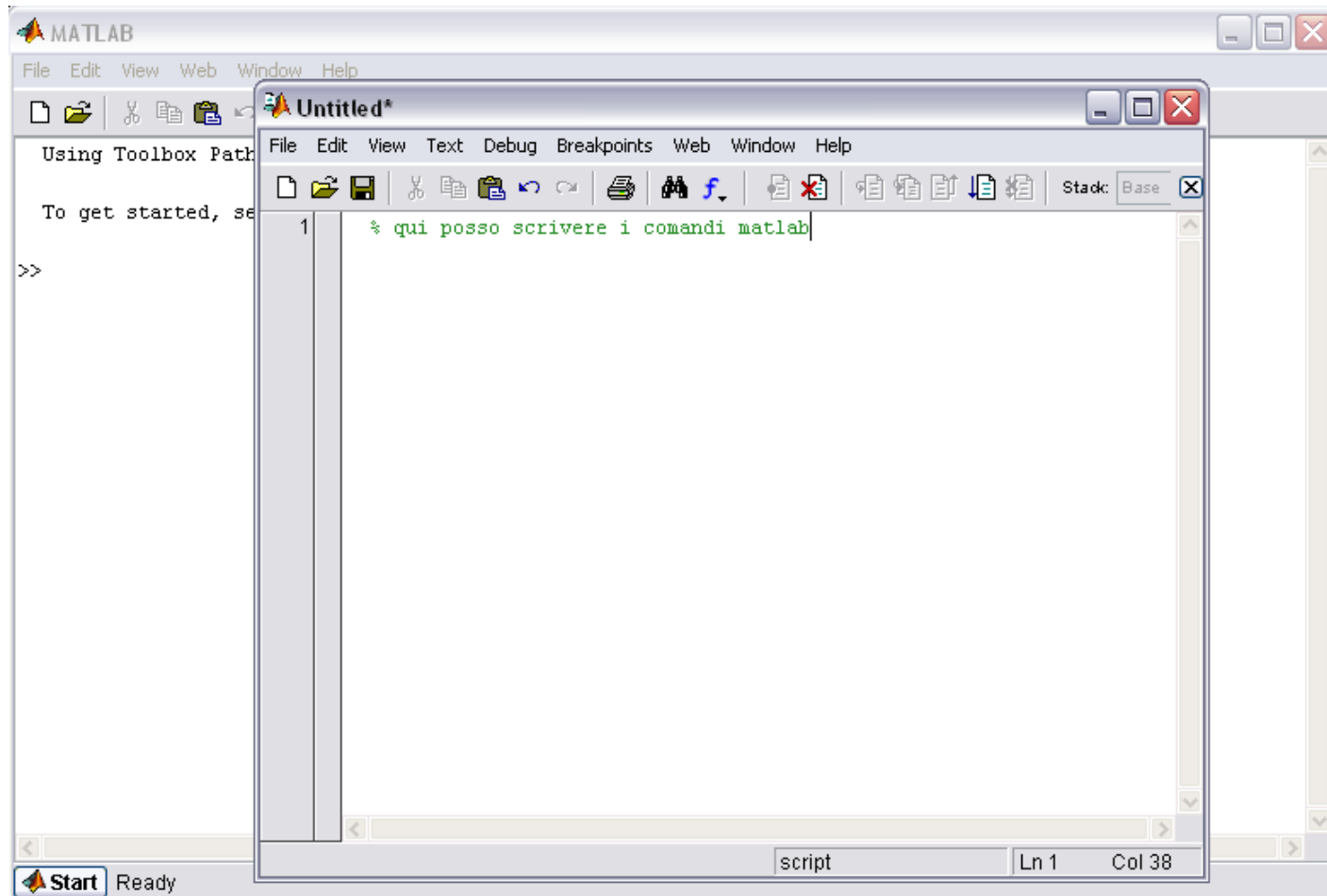
Interfaccia utente: la **Command Window** dà accesso all'interprete mediante scrittura diretta dei comandi al prompt **>>**



L'editor (1)

- Invece di inserire direttamente il codice al prompt di MATLAB è conveniente editarlo in una finestra separata: l' **Editor**
- Il file che contiene il codice può essere salvato come file di testo ed avrà estensione ***.m**
- Il codice può essere eseguito direttamente dall'editor ed eventualmente “debuggato”

L'editor (2)



Matlab come calcolatrice

- La modalità di impiego più semplice è quella per valutare espressioni numeriche
- **Esempio:** calcolare $4 + \sqrt{2} - \sin(0.2\pi)^2 + e^2$
 - » `x=4 + sqrt(2) - sin(0.2*pi)^2 + exp(2)`
x =
12.4578
- Il risultato viene scritto nella variabile **x**
- Aggiungendo `;` (punto e virgola) alla fine del comando si evita la visualizzazione del risultato

Definizione di variabili

- E' possibile calcolare espressioni algebriche in cui appaiono variabili già definite
- **Esempio**
 - » $a=4; b=2;$
 - » $c=a*b$
 - $c=$
 8
- Per cancellare una variabile (ad esempio **a**)
 - » `clear a`
- Per cancellare tutte le variabili
 - » `clear all`

II Workspace

- Ogni variabile definita in questo modo viene conservata in memoria, nel **Workspace**
- Il comando **whos** mostra una lista delle variabili definite, con dimensioni e tipo

» whos

Name	Size	Bytes	Class
a	1x1	8	double array
x	1x1	8	double array
b	1x1	8	double array

Grand total is 3 elements using 24 bytes

Lettura e scrittura su file

- mediante i comandi **load** e **save** è possibile salvare su file le variabili del workspace
 - `load nomefile var1 var2 ...`
carica dal file `nomefile.mat` le variabili elencate
 - `save nomefile var1 var2 ...`
scrive nel file `nomefile.mat` le variabili elencate
 - `load nomefile` carica tutte le variabili in `nomefile`
 - `save nomefile` salva tutto il workspace in `nomefile.mat`

Funzioni e variabili

- Esiste un insieme molto vasto di funzioni predefinite (come `sin`, `exp` e `sqrt` nell'esempio precedente)
- A differenza di altri linguaggi (C, Pascal, ...) non occorre `dichiarare` le variabili (l'assegnazione coincide con la dichiarazione)

Esempi di funzioni predefinite

- Funzioni trigonometriche (`sin`, `cos`, `tan`, `acos`, `asin`, `atan`, ...)
- Esponenziale e logaritmo (`exp`, `log`, `log10`, ...)
- Radice quadrata (`sqrt`)
- Numeri complessi (`abs` \rightarrow modulo, `angle` \rightarrow fase, `real` \rightarrow parte reale, `imag` \rightarrow parte immaginaria, ...)
- L'unità immaginaria è indicata con `i` oppure `j`

Esempi di uso di funzioni predefinite

» `x=abs(2+3*i)`

x =

3.6056

» `y=20*log10(200)`

y =

46.0206

» `z=sqrt(-4)`

z =

0 + 2.0000i

Un comando molto utile

help

- seguito dal nome di una funzione restituisce una sintetica descrizione e la sintassi d'uso
- “da solo” restituisce l'elenco di TUTTE le funzioni di Matlab, ordinate per categorie

Definizione di matrici

- Come si definisce una matrice in Matlab?

Esempio

» A=[1 , 2 ; 3 , 4]

A =

1 2
3 4

$$A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$$

- Come si accede agli elementi di una matrice?

» x=A(2,1)

x =

3

**indici (riga e colonna)
dell'elemento di interesse**

La wildcard :

- Per accedere a intere righe o colonne di una matrice, si usa la wildcard :

- Es.: selezionare la prima riga di A

» `x=A(1,:)`

x =

1 2

$$A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$$

- Es.: selezionare la seconda colonna di A

» `y=A(:,2)`

y =

2

4

Operazioni elementari sulle matrici (1)

- Sono definiti gli operatori $+$, $-$, $*$, $^{\wedge}$
- Esempio: prodotto di matrici

» $A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$;

» $B = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$;

» $C = A * B$

$C =$

9 12 15

19 26 33

- Esempio: potenza di matrice

» $A2 = A^{\wedge}2$

$A2 =$

7 10

15 22

Operazioni elementari sulle matrici (2)

- Matrice trasposta

» $A_{trasp} = A'$

$A_{trasp} =$

1 3

2 4

- Matrice inversa

» $A_{inv} = \text{inv}(A)$

$A_{inv} =$

-2.0000 1.0000

1.5000 -0.5000

Operazioni elementari sulle matrici (3)

- Dimensioni di una matrice

» `dimA=size(A)`

`dimA =`

`2 2`

- Matrice con elementi unitari

» `A1=ones(size(A))`

`A1 =`

`1 1`

`1 1`

Operazioni elementari sulle matrici (4)

- Matrice identità

» `Id=eye(size(A))`

Id =

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

- Matrice nulla

» `N=zeros(size(A))`

N =

$$\begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

Operazioni elementari sulle matrici (5)

- Determinante:

» $dA = \det(A)$

$dA =$

-2

- Traccia:

» $tA = \text{trace}(A)$

$tA =$

5

Alcune matrici “speciali”

- `eye(n)` – matrice identità $n \times n$
- `zeros(n,m)` – matrice di 0 $n \times m$
- `ones(n,m)` – matrice di 1 $n \times m$

Autovalori, autovettori e polinomio caratteristico

- Autovalori

» `lambda=eig(A)`

lambda =

-0.3723

5.3723

- Autovalori e autovettori

» `[M,D]=eig(A)`

M =

-0.8246 -0.4160

0.5658 -0.9094

D =

-0.3723 0

0 5.3723

- Polinomio caratteristico $\det(\lambda I - A) = \lambda^2 - 5\lambda - 2$

» `pc=poly(A)`

pc =

1.0000 -5.0000 -2.0000

(NOTA: il risultato fornisce i coefficienti del polinomio caratteristico)

Vettori

- I vettori hanno due funzioni fondamentali in Matlab:
 - rappresentazione dei **polinomi** (un polinomio è descritto dal vettore dei suoi coefficienti)
(Come appena visto per il polinomio caratteristico di A)
 - rappresentazione di **segnali** (un segnale è rappresentato mediante la sequenza dei valori che assume in un insieme di istanti di tempo)

Definizione di vettori (1)

» `v=0:10`

`v =`

0 1 2 3 4 5 6 7 8 9 10

» `v=1:0.5:3`

`v =`

1.0000 1.5000 2.0000 2.5000 3.0000

passo

valore iniziale

valore finale

Definizione di vettori (2)

- Come matrici riga o colonna

» $v = [3 \ 6 \ 1 \ 7]$

$v =$

3 6 1 7

» $w = [3 \ 6 \ 1 \ 7]'$

$w =$

3

6

1

7

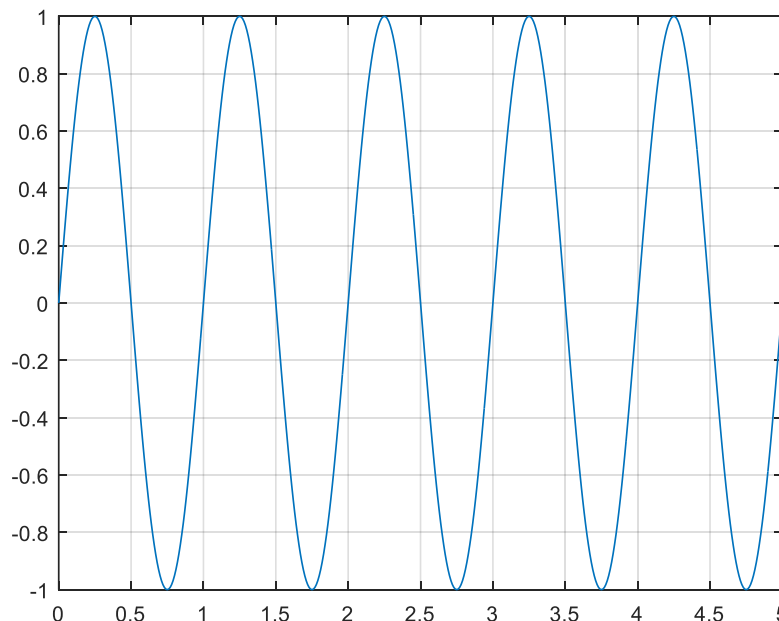
Rappresentazioni grafiche

`plot(x,y)` traccia il grafico dei punti che hanno come ascisse e come ordinate gli elementi dei vettori `x` e `y`, rispettivamente

- Esempio

- » `x=0:0.01:5;`
- » `y=sin(2*pi*x);`
- » `plot(x,y);`
- » `grid;`

griglia



Sistemi dinamici lineari

- Un sistema dinamico lineare può essere descritto in forma di **variabili di stato** mediante il comando **ss**
 - a tempo continuo
 - » `sistema=ss(A,b,c,d);`
 - a tempo discreto
 - » `sistema=ss(A,b,c,d,1);`



Passo unitario

Esempio

- Definizione del sistema

$$\dot{x}(t) = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} x(t) + \begin{bmatrix} 1 \\ 0 \end{bmatrix} u(t)$$

$$y(t) = \begin{bmatrix} 1 & 1 \end{bmatrix} x(t)$$

```
» A=[1 2 ; 3 4];  
» B=[1 0]';  
» C=[1 1];  
» sistema=ss(A,B,C,0);
```

Simulazione di sistemi lineari

- Alcune funzioni disponibili per la simulazione
 - **initial** -> simulazione movimento libero
 - **lsim** -> simulazione con ingresso qualsiasi e stato iniziale qualsiasi

- Sintassi

- » `[y,t,x]=initial(sistema);`

- » `[y,t,x]=lsim(sistema,u,t);`

con c.i. nulla

vettore dei tempi

vettore sequenza ingresso

- » `[y,t,x]=lsim(sistema,u,t,xin);`

con c.i. da specificare come vettore riga in xin

Esempio

```
» A=[-1 10 ; -10 -1];  
» b=[1 0]';  
» c=[1 1];  
» sistema=ss(A,b,c,0);  
» t=(0:0.01:10);  
» u=3*ones(size(t));  
» y=lsim(sistema,u,t);  
» plot(t,y);  
» xlabel('tempo t');  
» ylabel('uscita y');
```

