



Introduzione a MATLAB

Laboratorio 1 - Segnali per le
Telecomunicazioni

Prof. Prati Claudio Maria

Autore: Federico Borra
Politecnico di Milano, DEIB
Email: federico.borra@polimi.it
Marzo '17, Ultima revisione: 21/03/17



- ❑ MATLAB (*MAT*rix *LAB*oratory) è un linguaggio di programmazione e un ambiente per il calcolo numerico
- ❑ MATLAB permette di manipolare matrici, plottare funzioni e dati, implementare algoritmi etc...
- ❑ MATLAB è molto usato in ambiente accademico al fine di sviluppare e testare algoritmi



- ❑ Come studenti del Politecnico di Milano avete una licenza gratuita annuale dei prodotti MathWorks
- ❑ Trovate maggiori dettagli per installare MATLAB a questo link:

<http://www.software.polimi.it/software-download/studenti/matlab/>

- ❑ MATLAB permette di creare e inizializzare variabili molto facilmente. La dichiarazione di variabili in MATLAB segue le seguenti regole:
 - tutte le variabili sono **matrici**;
 - **non** si dichiara il **tipo** di variabile

```
>> a=5           % variabile scalare (1x1)
>> b=[ 4  6 ]    % vettore riga (1x2)
>> c=[ -5;  2 ]  % vettore colonna (2x1)
>> d=[ 2, 3; -1, 7 ] % matrice quadrata (2x2)
```

- ❑ Negli esempi precedenti abbiamo utilizzato gli operatori:
 - separatore di riga: punto e virgola (;)
 - separatore di colonna: virgola (,) o spazio bianco



- MATLAB può effettuare varie operazioni tra matrici. Esse possono essere raggruppare in due categorie:
 - Operazioni matriciali
 - Operazioni elemento per elemento

- ❑ Le principali operazioni matriciali sono definite nel seguente modo

```
>> C=A+B           % somma di matrici
>> C=A*B           % prodotto di matrici
>> C=A/B           % divisione tra matrici C=AB-1
>> C=A^3           % elevamento a potenza di una
                   % matrice C=A*A*A
```

- ❑ **ATTENZIONE:** queste operazioni sono ben definite soltanto se le matrici hanno *dimensioni compatibili*.



- ❑ Per eseguire operazioni tra matrici elemento per elemento bisogna *aggiungere un punto* davanti all'operatore

```
>> C=A.*B           % prodotto elemento per elemento
>> C=A./B           % divisione elemento per elemento
>> C=A.^3           % elevamento a potenza elemento per
                    elemento
```

- ❑ Osservazione: nei primi due casi (prodotto e divisione elemento per elemento) A e B devono avere le stesse dimensioni

- ❑ Esempio (moltiplicazione elemento per elemento):

$$\begin{bmatrix} 1 & 3 \\ 2 & 2 \end{bmatrix} \cdot * \begin{bmatrix} 2 & 1 \\ 3 & 2 \end{bmatrix} = \begin{bmatrix} 2 & 3 \\ 6 & 4 \end{bmatrix}$$



□ Definizione di vettori

```
>> x=[0:0.1:1]           % vettore di punti tra 0 e 1
                           % equispaziati di 0.1
>> y=linspace(0,1,10)    % vettore di 10 elementi
                           % equispaziati tra 0 e 1
```

□ Estrazione di sotto-matrici

```
>> A(2,3)                % estrae l'elemento  $A_{23}$ 
>> A(:,3)                % estrarre la terza colonna di A
>> A(1:4,3)              % estrarre le righe da 1 a quattro
                           % della terza colonna
>> A(1,:)                % estrae la prima riga di A
>> diag(A)               % estrae gli elementi sulla diagonale
                           % di A
```




□ Costruzione di matrici particolari

```
>> A=eye(n)           % matrice identità n x n
>> A=zeros(n,m)      % matrice n x m di soli zeri
>> A=ones(n,m)       % matrice n x m di soli uni
>> A=rand(n,m)       % matrice aleatoria n x m
>> A=diag(v)         % matrice avente v come diagonale
```

□ Funzioni matriciali

```
>> C=A'              % trasposta di A
>> C=inv(A)          % inversa di A
>> d=det(A)          % determinante di A
>> r=rank(A)         % rango di A
>> v=eig(A)          % autovalori e autovettori di A
>> v=cond(A)         % numero di condizionamento di A
```



□ Ciclo FOR

```
%% Prodotto scalare tra due vettori
>> ps = 0;
>> for i = 1:n
>>     ps = ps + x(i)*y(i);
>> end
%% Si può fare in modo molto più efficiente così
>> ps=x'*y;
```

□ Ciclo WHILE

```
%% Prodotto scalare tra due vettori
>> ps = 0;
>> i = 0;
>> while (i < n)
>>     i = i + 1;
>>     ps = ps + x(i)*y(i);
>> end
```



□ Istruzione condizionale IF

```
>> if (r >= 0)
>>     radice = sqrt(r);
>> end
```

□ Operatori logici a disposizione

operatore	azione logica
&	<i>and</i>
	<i>or</i>
~	<i>not</i>
==	<i>equal to</i>



- Il comando Help è il modo più semplice per determinare la sintassi ed il comportamento di una particolare funzione. Informazioni sono esposte direttamente nella finestra di comando. Per esempio:

```
>> help inv
inv Matrix inverse.
inv(X) is the inverse of the square matrix X.
A warning message is printed if X is badly scaled
or nearly singular.
```



- ❑ Uno script è un file di testo contenente una sequenza di comandi MATLAB che viene salvato con estensione .m

- ❑ I comandi all'interno di uno script sono eseguiti sequenzialmente, come se fossero scritti nella finestra dei comandi

- ❑ Uno script può
 - Essere ri-eseguito
 - Essere facilmente modificato

- ❑ Uno script opera sulle variabili del workspace che può arricchire introducendone di nuove



Numeri Complessi

Laboratorio 1 - Segnali per le
Telecomunicazioni

Prof. Prati Claudio Maria

Autore: Federico Borra
Politecnico di Milano, DEIB
Email: federico.borra@polimi.it
Marzo '17, Ultima revisione: 21/03/17



$$x = a + ib$$

□ Parte reale : $Re[x] = a$

□ Parte immaginaria: $Im[x] = b$

□ Modulo: $\|x\| = \sqrt{a^2 + b^2}$

□ Fase: $\angle x = \text{atan2}(b, a)$

□ Coniugato: $x^* = a - ib$



$$Ae^{i\phi} = A \cos(\phi) + iA \sin(\phi)$$

□ A = modulo

□ ϕ = fase

□ Alcuni esempi:

➤ $e^{i0} = \cos(0) + i \sin(0) = 1$

➤ $e^{i\pi/2} = \cos(\pi/2) + i \sin(\pi/2) = i$

➤ $e^{i\pi} = \cos(\pi) + i \sin(\pi) = -1$

➤ $e^{i3\pi/2} = \cos(3\pi/2) + i \sin(3\pi/2) = -i$



$$A[n]e^{i\phi[n]} = A[n] \cos(\phi[n]) + iA[n] \sin(\phi[n])$$

- Parte reale : $A[n] \cos(\phi[n])$
- Parte immaginaria: $A[n] \sin(\phi[n])$
- Modulo: $A[n]$
- Fase: $\phi[n]$

$$A[n]e^{i\phi[n]} = A[n] \cos(\phi[n]) + iA[n] \sin(\phi[n])$$

□ Esempi:

➤ $A[n] = 1 \quad \forall n$

➤ $\phi[n] = \alpha n$

➤ Oppure

➤ $A[n] = e^{-\beta n}$

➤ $\phi[n] = \alpha n$

➤

$$A[n]e^{i\phi[n]} = A[n] \cos(\phi[n]) + iA[n] \sin(\phi[n])$$

- In MATLAB questa funzione è vista come un vettore che contiene i campioni per ogni n definito per un intervallo

$n=0$	$n=1$	$n=2$
$A[0]e^{i\phi[0]}$	$A[1]e^{i\phi[1]}$	$A[2]e^{i\phi[2]}$

□ Come vengono definite per una funzione reale la parte reale, la parte immaginaria, il modulo e la fase?

□ Vediamo un esempio:

$$f(x) = 2 \cos(x)$$

□ Parte reale : $2 \cos(x)$

□ Parte immaginaria: $0 \quad \forall x$

□ Modulo: $|2 \cos(x)|$

□ Fase: $\begin{cases} 0, & \text{if } f(x) \geq 0 \\ \pi, & \text{if } f(x) < 0 \end{cases}$ (Vedi slide seguente)

- Ricordiamo che un **numero positivo** α può essere riscritto in forma di esponenziale complesso nel seguente modo

$$\alpha = \alpha e^{i0} \quad \alpha \geq 0$$

- Mentre un **numero negativo** β nel seguente modo

$$\beta = |\beta| e^{i\pi} \quad \beta < 0$$