

UNIVERSITA' DEGLI STUDI DI PAVIA

FACOLTA' DI INGEGNERIA

Corso di Fondamenti di Automatica

Prof. D.M. Raimondo, Prof. L. Magni, Prof. C. Toffanin, Prof. Cucuzzella

Introduzione a MATLAB



Accesso Account

E' possibile loggarsi alle postazioni in C2-C3 tramite le seguenti credenziali:

- 1. Username: c123456 (123456 rappresenta il numero di matricola dello studente)**
- 2. Password: inf56 (56 rappresentano gli ultimi due numeri della matricola)**
- 3. Gli account dovrebbero già esistere dal primo anno (stesse credenziali usate per il corso di Fondamenti di Informatica)**
- 4. Per eventuali problemi di account recarsi in C4**

Cos' è MATLAB

MATLAB è un programma interattivo che utilizza un linguaggio di alto livello e si è imposto in ambito ingegneristico mondiale come efficace strumento di calcolo e simulazione.

Il nome deriva da **MAT**rix **LAB**oratory: l'elemento base è la matrice.

Il software è corredato di una famiglia di soluzioni specifiche per varie applicazioni denominate Toolbox. Si tratta di insiemi di funzioni **MATLAB** aggiuntive che estendono l'ambiente di lavoro per risolvere particolari categorie di problemi quali ad esempio sistemi di controllo, matematica finanziaria, acquisizione e elaborazione di immagini...

Cos' è MATLAB

E' possibile scaricare Matlab gratuitamente come presentato sulla pagina:

<http://news.unipv.it/?p=16675>

1. **Creare un Account MathWorks usando il proprio indirizzo e-mail universitario**
2. **Associare la Licenza usando il Codice di Attivazione di unipv**
3. **Fare il download per la release attuale.**
4. **Installare la versione**
5. **Attivare Matlab tramite l'utente appena creato**
6. **Toolbox fondamentali: Control System Toolbox e Simulink (consigliati per usi futuri System Identification Toolbox)**

Cos' è MATLAB

Il modo più elementare per usare **MATLAB** è quello di utilizzarlo come una calcolatrice per fare operazioni quali somme, sottrazioni, moltiplicazioni, divisioni e potenze (+, -, *, /, ^).

```
>> (4*10+8)/3
```

Il punto e virgola ; al termine di un'istruzione impedisce la visualizzazione del risultato dell'istruzione stessa.

MATLAB è un interprete di comandi che vengono eseguiti nell'ordine in cui sono impartiti.

Variabili ed espressioni

Le istruzioni MATLAB sono spesso nella forma

variabile = espressione

o semplicemente

espressione

Le espressioni sono costituite da operatori (+ * - / ^) , caratteri speciali, funzioni, variabili e numeri

- funzioni: nomi simbolici con argomenti fra parentesi: `eye(2,2)`
- numeri: reali, ad es. 5, e complessi, ad es. `5 + 2*i` o indifferentemente `5 + 2*j`

Quando una variabile viene definita, essa viene memorizzata nel *workspace*.

Esercizio 1: Matrici

Introdurre le matrici $A = \begin{pmatrix} -1 & 3 & 2 \\ 5 & -8 & 6 \\ 1 & 0 & 1 \end{pmatrix}$ $B = \begin{pmatrix} 1 & 2 & 6 \\ 3 & 4 & 0 \\ -1 & -1 & 2 \end{pmatrix}$

e calcolare: A' , $A+B$, $A-B$, $4*A$, $A*B$, A^2 .

Autovalori e autovettori

Data la matrice quadrata A di dimensioni $n \times n$ l'istruzione

$a = \text{eig}(A)$

(“eig”=eigenvalues) restituisce il vettore colonna degli autovalori della matrice A .

$[V, D] = \text{eig}(A)$

restituisce la matrice V $n \times n$ degli autovettori normalizzati e la matrice diagonale D $n \times n$, che presenta sulla diagonale gli autovalori della matrice A .

Altre funzioni consentono di calcolare il determinante $\text{det}(A)$, l'inversa $\text{inv}(A)$ e il rango $\text{rank}(A)$

I comandi più utili!

help richiama l' help in linea.

help <nome istruzione> richiama l' help per una specifica istruzione

doc finestra della documentazione (solo nelle versioni più recenti di MatLab)

... intere generazioni di ricercatori hanno imparato MatLab analizzando m-files e imparando nuove istruzioni con il comando help !

Ancora sulle matrici

- A(i, j)** seleziona l'elemento di posto (i,j),
- A(:, j)** seleziona tutta la colonna j-esima di A,
- A(i, :)** seleziona tutta la riga i-esima di A,
- A(:, end)** seleziona tutta l'ultima colonna di A,
- A(end, :)** seleziona tutta l'ultima riga di A,
- A=[]** crea una matrice A vuota o cancella il suo contenuto,
- v(i)** seleziona l'elemento i-esimo del vettore v.
- [m, n]=size(A)** numero di righe (m) e colonne (n) di A
- m=length(v)** numero di elementi del vettore v

Ancora sui vettori

Esistono vari modi per generare dei vettori

```
>> x=1:5
```

```
x =
```

```
1 2 3 4 5
```

```
>> x=1:0.5:2
```

```
x =
```

```
1.0000 1.5000 2.0000
```

Polinomi

Un polinomio è rappresentato da un vettore riga che contiene i coefficienti del polinomio in ordine decrescente delle potenze del polinomio.

Ad esempio

$p = [1 \ 0 \ -6 \ 3]$

permette di rappresentare $x^3 - 6x + 3$.

$r = \text{roots}(p)$ permette di trovare le radici di p .

Polinomi

Il prodotto di due polinomi $a(x)$ e $b(x)$ si ottiene effettuando la convoluzione dei loro coefficienti ($\text{conv}(a,b)$). Ad esempio

$$a(x) = x^2 + 2x + 3 \quad \text{e} \quad b(x) = 4x^2 + 5x + 6$$

$$a(x) * b(x) = 4x^4 + 13x^3 + 28x^2 + 27x + 18$$

si ottiene mediante

```
>> a=[1 2 3]; b=[4 5 6];
```

```
>> c=conv(a,b)
```

```
c =
```

```
4 13 28 27 18
```

Funzioni matematiche

Seno	<code>sin (z)</code>	<code>sind (z)</code>
Coseno	<code>cos (z)</code>	<code>cosd (z)</code>
Tangente	<code>tan (z)</code>	<code>tand (z)</code>
Arcotangente	<code>atan (y)</code>	<code>atand (y)</code>
Esponenziale	<code>exp (x)</code>	
Logaritmo naturale	<code>log (x)</code>	
Logaritmo in base 10	<code>log10 (x)</code>	
Radice quadrata	<code>sqrt (x)</code>	

Se gli argomenti sono vettori, le funzioni vengono applicate ad ogni singolo elemento (e restituiscono un vettore)

Operazioni elemento per elemento

Sono precedute da un punto : `.*` `.^` `./`

Se applicate a vettori (o matrici) agiscono su ogni elemento del vettore

```
>> x=1:5
```

```
x =
```

```
1 2 3 4 5
```

```
>> y=x.^2
```

```
y =
```

```
1 4 9 16 25
```

Grafici

- l'istruzione `plot` consente di effettuare grafici bidimensionali.

`plot(x, y)`

con `x` e `y` vettori $1 \times n$ produce un grafico con `x` in ascissa e `y` in ordinata;

Esercizio: si tracci il grafico di $y = x^2$ fra -10 e 10.

- l'istruzione `grid` visualizza una griglia sullo schermo;
- le istruzioni `title('titolo')`, `xlabel('label')` e `ylabel('label')` consentono di mettere il titolo e delle etichette sull'asse `x` e `y`, rispettivamente;
- l'istruzione `axis([xinit xend yinit yend])` consente di riscalare il grafico.

Esercizio 2

Disegnare il grafico di $y = \sin(x)$ e di $y = \cos(x)$ fra 0 e 2π sulla stessa finestra in modo da poterli confrontare.

Per evitare la cancellazione di un grafico da parte di quello successivo, fra i diversi comandi `plot` va interposto il comando `hold on`.

Se invece si desidera tracciare i grafici in differenti figure è necessario premettere all'istruzione `plot` il comando `figure(n)` che apre la figura numero n (o la seleziona se è già aperta).

Per cancellare la figura corrente: `clf`

Modifica del tipo e del colore della linea

È possibile far tracciare il grafico con linee di colori e tratti differenti da quelli di default usando la sintassi

```
plot(x,y,'colore_linea_simbolo')
```

'colore_linea_simbolo' è una particolare stringa formata da uno a quattro caratteri in combinazione fra quelli riportati in tabella.

Simbolo		Colore		Linea	
+	più	c	ciano	-	linea continua
o	circoletto	m	magenta	--	linea tratteggiata
*	asterisco	y	giallo	:	linea punteggiata
x	per	r	rosso	-.	linea tratto punto
s	quadrato	g	verde		
d	diamante	b	blu		
^	triangolo	w	bianco		
v	triangolo	k	nero		
>	triangolo				
<	triangolo				
'p'	pentagono				
'h'	esagono				

M-files

I files che contengono istruzioni eseguibili da MATLAB sono detti m-files in quanto hanno come suffisso .m

- Se un m-file memorizza una sequenza di istruzioni è detto scriptfile.

Se contiene una nuova funzione viene detto function file.

- Un function file:

```
function y=linear2(alfa,beta,x)
% questo file restituisce una
% trasformazione affine del vettore x;
y= alfa + beta *x;
```

```
>> x=[1 4 5 9];
```

```
>> linear2(1,4,x)
```

```
ans =
```

```
5 17 21 37
```

Cicli

MATLAB possiede le usuali istruzioni di controllo che consentono di effettuare programmazione:

for condizione ...istruzioni ... **end**

while condizione ...istruzioni ... **end**

if condizione ...istruzioni ... **else** ... istruzioni **end**

```
% script file sommatoria.m Calcola la somma
% a1+a2+ ... + a9 ove ai=i^2/(i+1)
s=0;
for i=1:9
    s=s+i^2/(i+1);
end
s
```

```
>> sommatoria
```

```
s =
37.9290
```

Interpretazione dei comandi MatLab

Quando l'interprete di MATLAB trova un nome, ad esempio "prova", segue, nell'ordine, i seguenti passi:

1. cerca nel workspace la variabile di nome prova
2. cerca una funzione built-in di nome prova
3. cerca un m-file di nome prova nella directory da cui si è lanciato MATLAB
4. cerca un m-file di nome prova nel path indicato in una speciale variabile di sistema detta MATLABPATH. La variabile MATLABPATH può essere aggiornata (il modo dipende dal sistema operativo e dalla versione di MatLab)

Comandi utili

who visualizza l'elenco delle variabili presenti nel workspace

clear cancella tutte le variabili in memoria,

clear nome_variabile cancella la variabile indicata,

save nome_file salva le variabili in memoria e i dati in esse contenuti nel file specificato,

load nome_file richiama in memoria le variabili e i dati contenuti nel file specificato,

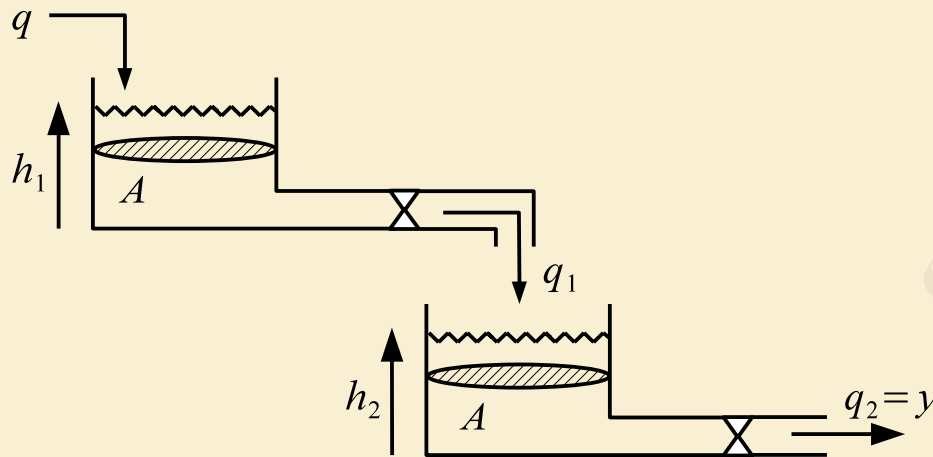
cd permette di cambiare directory,

dir o **ls** esamina il contenuto della directory corrente

Control System Toolbox

È un Toolbox di MATLAB per la modellizzazione, la simulazione e l'analisi dei sistemi dinamici.

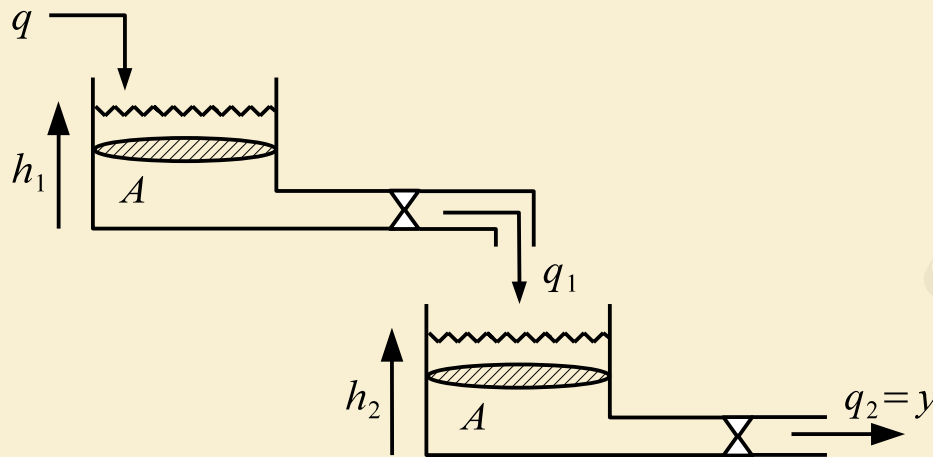
Esempio. Si consideri il seguente sistema idraulico



Control System Toolbox

È un Toolbox di MATLAB per la modellizzazione, la simulazione e l'analisi dei sistemi dinamici.

Esempio. Si consideri il seguente sistema idraulico



Parametri

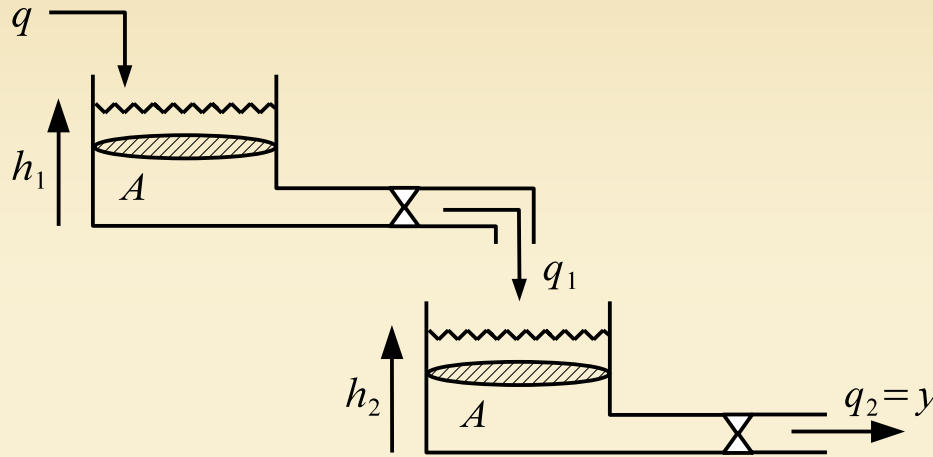
$$A = 1 \text{ m}^2$$

$$q_1 = 3 h_1 [\text{m}^3 / \text{s}]$$

$$q_2 = 5 h_2 [\text{m}^3 / \text{s}]$$

Simulare l'andamento dell'uscita $y(t), t \in [0,5]$ quando le altezze iniziali di acqua nei serbatoi sono $h_1(0) = 0.2 \text{ m}$, $h_2(0) = 0.1 \text{ m}$ e si immette la portata di ingresso $q(t) = \text{sen}(5\pi t) + 2$

Control System Toolbox



Parametri

$$A = 1 \text{ m}^2$$

$$q_1 = 3 h_1 [\text{m}^3 / \text{s}]$$

$$q_2 = 5 h_2 [\text{m}^3 / \text{s}]$$

Modello

$$A \dot{h}_1 = q - 3h_1$$

$$A \dot{h}_2 = 3h_1 - 5h_2$$

$$y = 5h_2$$

$$x_1 = h_1, x_2 = h_2, u = q$$

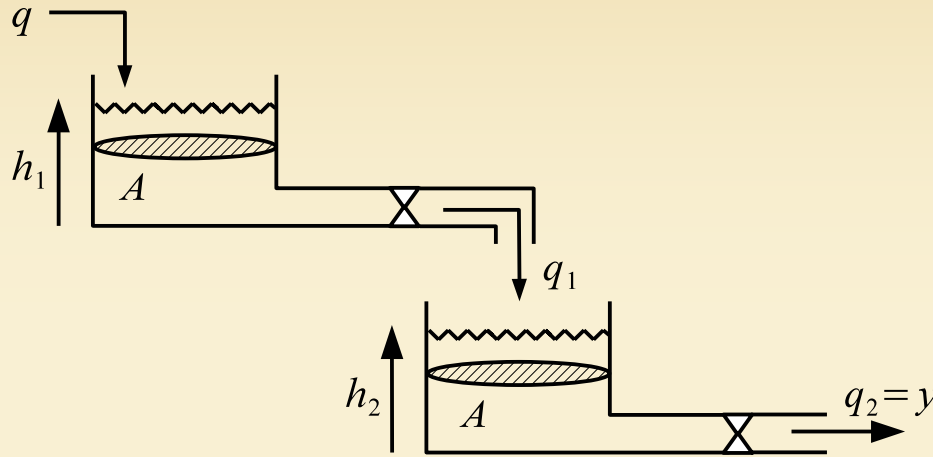


Sistema dinamico

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} -3 & 0 \\ 3 & -5 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} u$$

$$y = \begin{bmatrix} 0 & 5 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

Control System Toolbox



Sistema dinamico

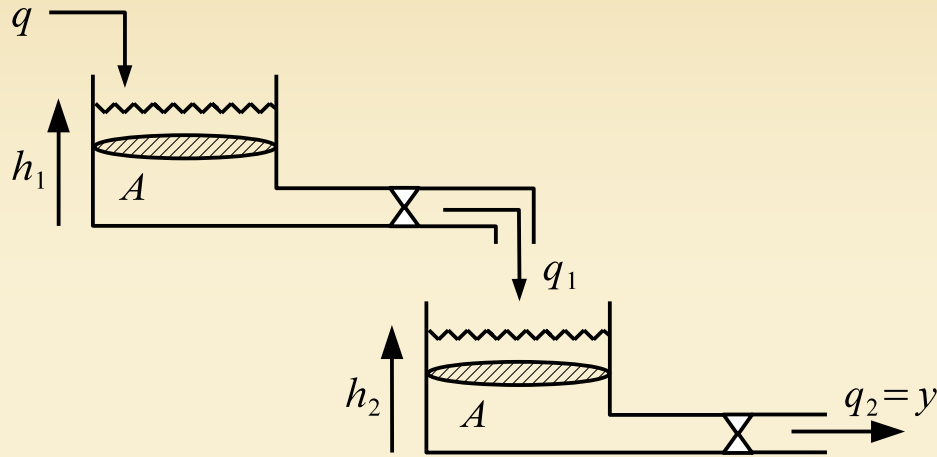
$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} -3 & 0 \\ 3 & -5 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} u$$
$$y = \begin{bmatrix} 0 & 5 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

Creare il seguente scriptfile "vasche.m"

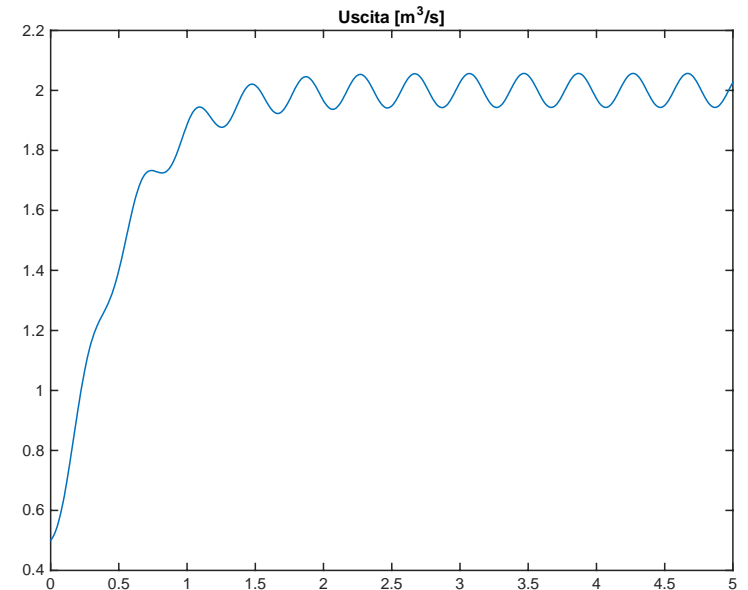
```
% Creazione di un oggetto  
%"state space"  
A=[-3 0; 3 -5];  
B=[1 0]';  
C=[0 5];  
D=0;  
sist=ss(A,B,C,D);
```

```
% Simulazione  
x0=[0.2 0.1]';  
tu=0:0.01:5;  
u=sin(5*pi*tu)+2;  
[y,t,x]=lsim(sist,u,tu,x0);  
  
% Visualizzazione  
plot(t,y)  
title('Uscita [m^3/s]')
```

Control System Toolbox



digitare **vasche** dal
prompt **>>** di MATLAB



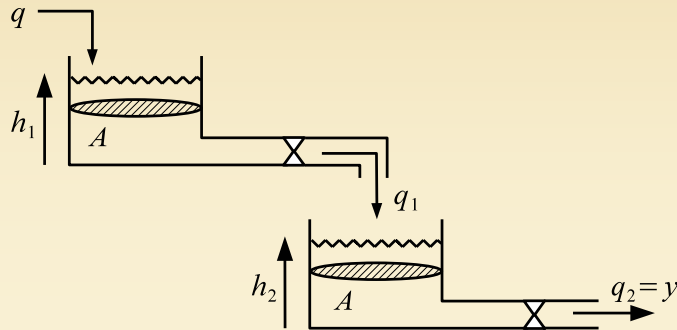
Simulink

Simulink (acronimo di **Simulation** e **Link**) è un Toolbox di **MATLAB** per la simulazione dei sistemi dinamici.

Attraverso un'interfaccia grafica è possibile creare modelli sotto forma di diagrammi a blocchi con pochi clic di mouse, grazie a una ricca libreria di componenti predefiniti (segnali di ingresso, componenti lineari e non lineari, connettori e uscite).

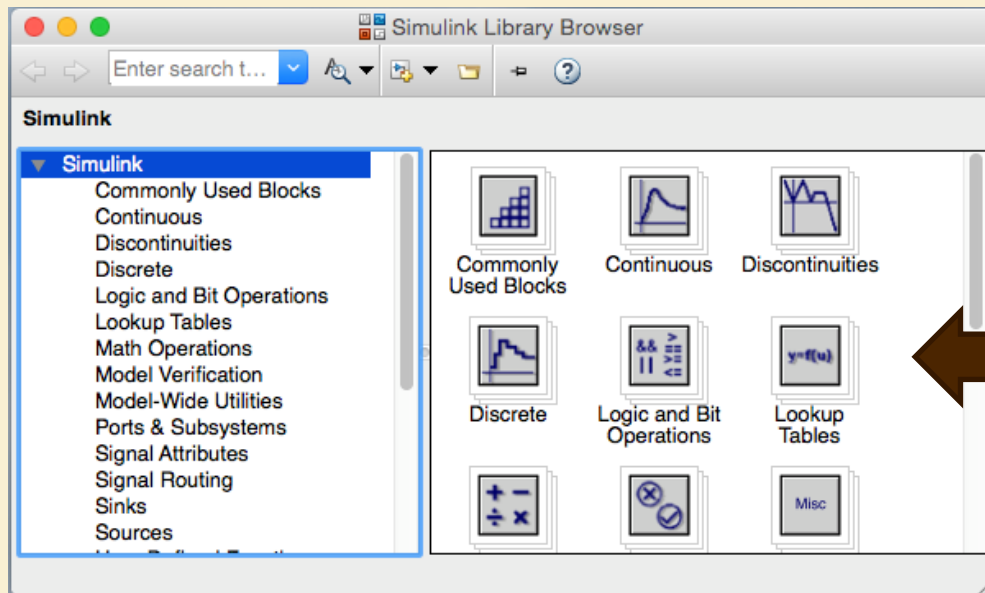
Dopo aver creato un modello è possibile eseguire simulazioni, modificando vari parametri, e vederne i risultati in tempo reale oppure memorizzarli nel **Workspace** di **MATLAB** per poterli rielaborare.

Simulink



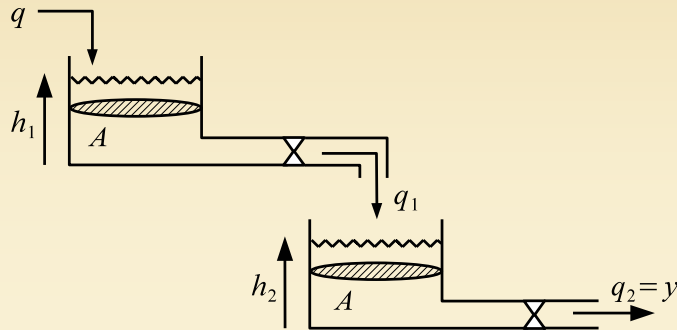
Simulare lo stesso
esperimento di prima
utilizzando Simulink

Per avviare Simulink : digitare **simulink** al prompt **>>** di MATLAB e premere <invio>.



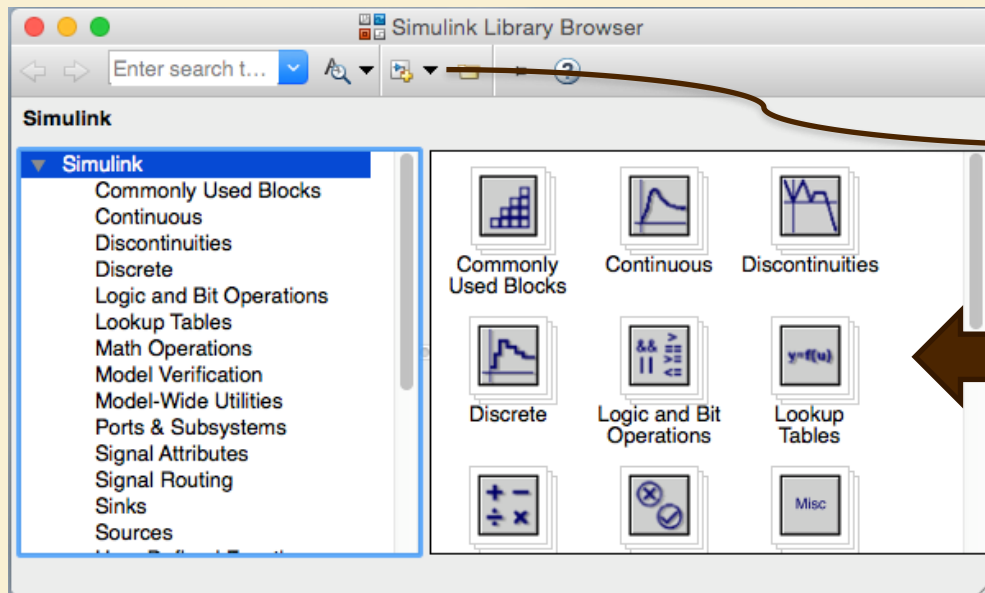
Librerie di blocchi

Simulink



Simulare lo stesso esperimento di prima utilizzando Simulink

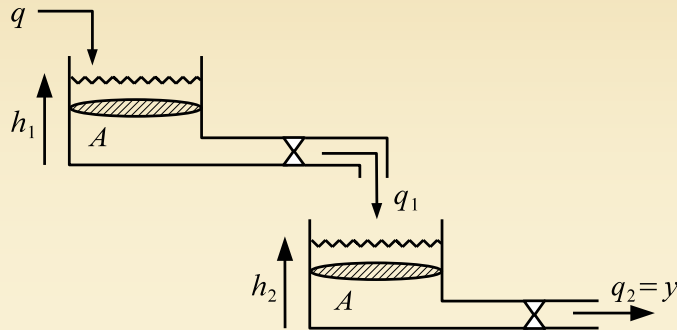
Per avviare Simulink : digitare **simulink** al prompt **>>** di MATLAB e premere <invio>.



Selezionare “new model”

Librerie di blocchi

Simulink

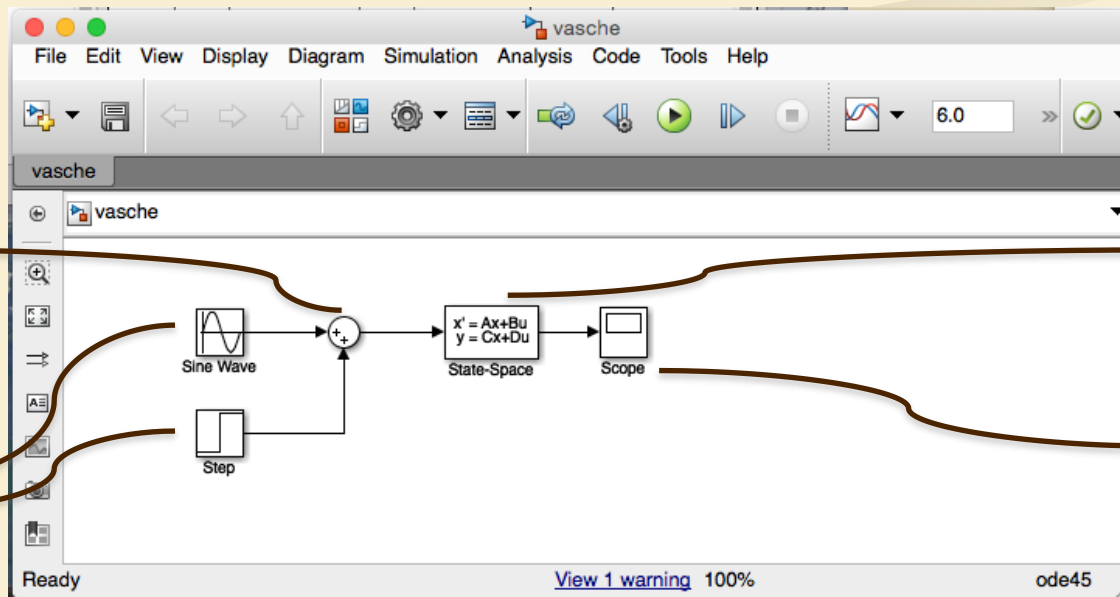


Simulare lo stesso
esperimento di prima
utilizzando Simulink

Trascinare i blocchi dalle librerie nel nuovo modello in modo da creare questo schema

“Math
operations”

“Sources”

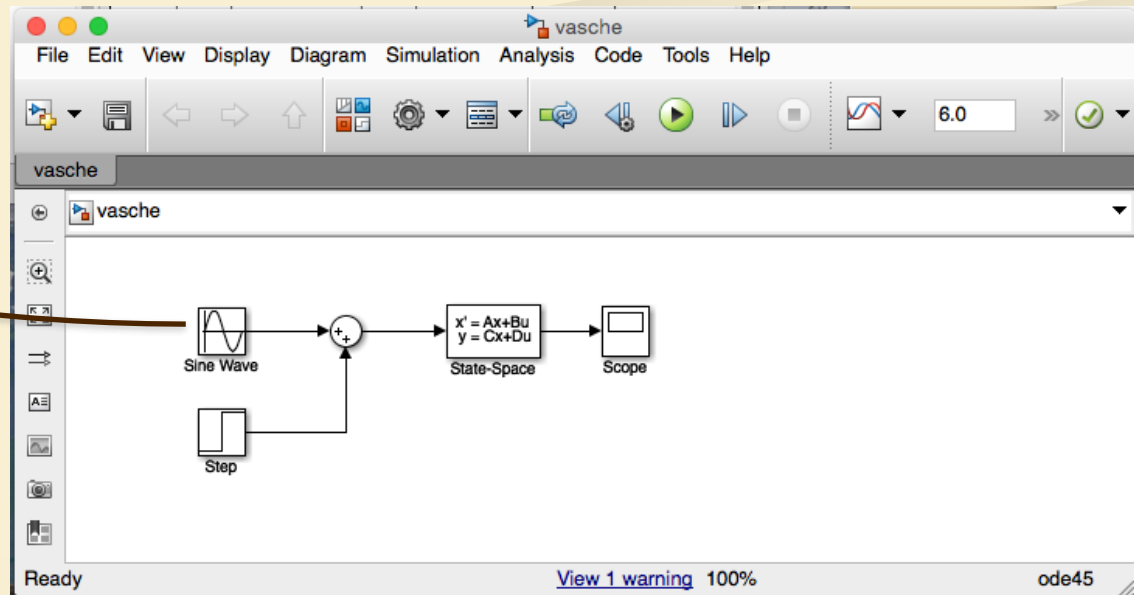
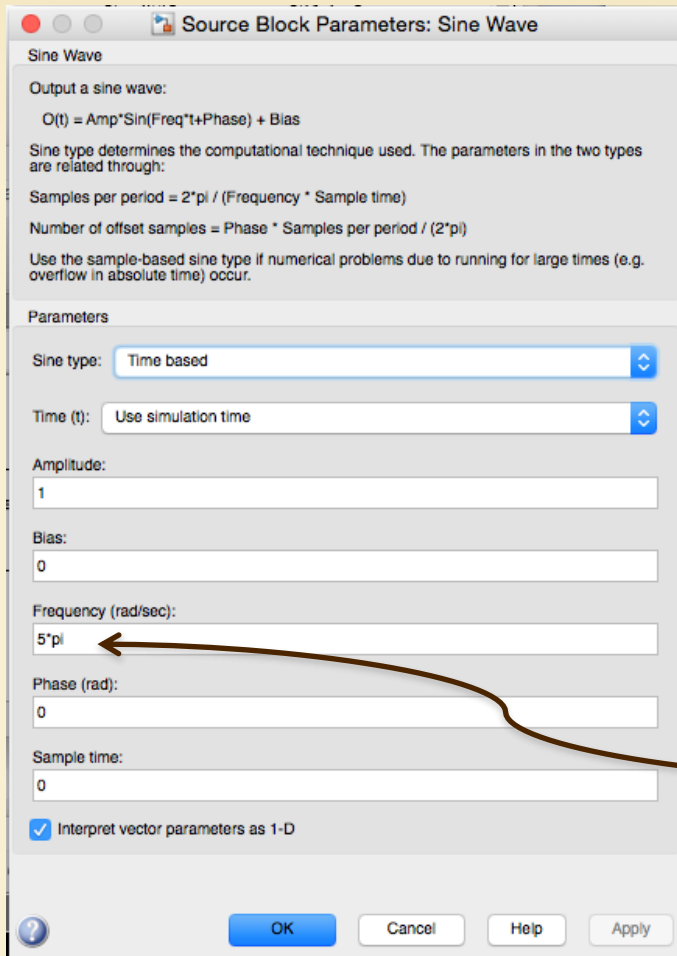


“Continuous”

“Sinks”

Simulink

Per modificare i parametri di un blocco, aprirne la maschera con doppio click



Simulink

Source Block Parameters: Step

Step

Output a step.

Parameters

Step time:
0

Initial value:
0

Final value:
2

Sample time:
0

Interpret vector parameters as 1-D

Enable zero-crossing detection

OK Cancel Help Apply

Function Block Parameters: State-Space

State Space

State-space model:
 $dx/dt = Ax + Bu$
 $y = Cx + Du$

Parameters

A:
A

B:
B

C:
C

D:
D

Initial conditions:
x0

Absolute tolerance:
auto

State Name: (e.g., 'position')
=

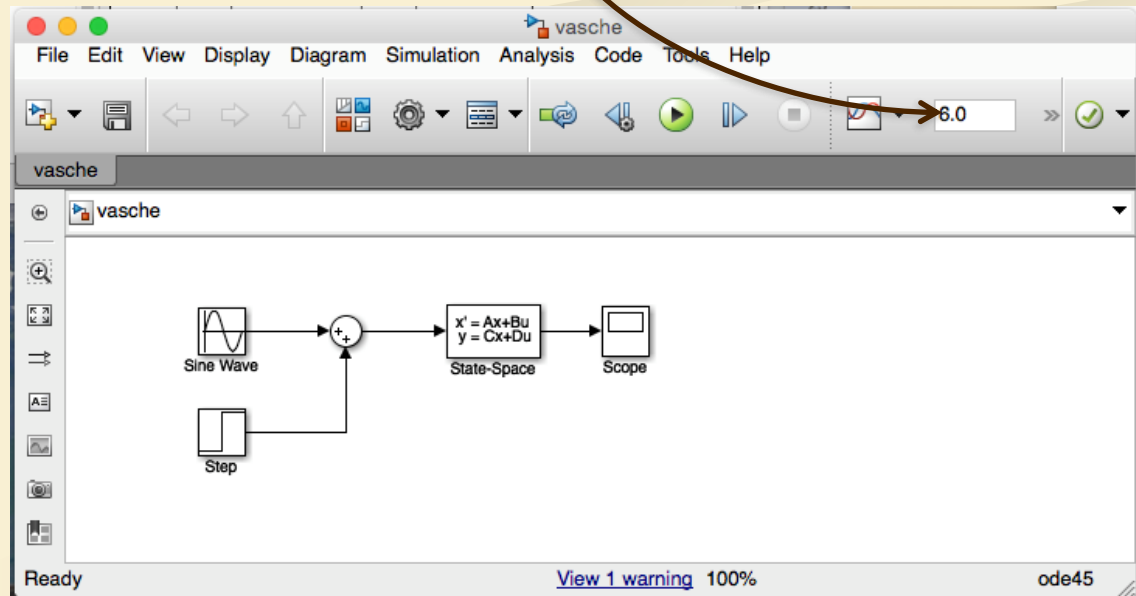
OK Cancel Help Apply

Ready View 1 warning 100% ode45

Simulink

Impostare il tempo di simulazione a 6 sec

Simulation -> Configuration parameters -> Stop time



Simulink

Eseguire la simulazione

