Automatica (Laboratorio)

Silvio Simani

Dipartimento di Ingegneria
Università di Ferrara
Tel. 0532 293844
Fax. 0532 768602

E-mail: ssimani@ing.unife.it

URL: http://www.ing.unife.it/~simani

URL: http://www.ing.unife.it/~simani/lessons.html

Automatica (Laboratorio)

Schema delle lezioni

1. Informazioni generali sul corso
2. Introduzione a Matlab
3. Simulazione di Sistemi Dinamici

⇛ **Introduzione a Simulink**

4. Osservatori e retroazione uscita-stato-ingresso
5. Modelli approssimati di sistemi dinamici
6. Progetto di Reti Correttrici
7. Identificazione di Sistemi Dinamici
8. Sintonizzazione di Controllori PID
9. Analisi di Sistemi a Dati Campionati

Bibliografia

⇒ Dispense del Corso di Laboratorio di Automatica. Sergio Beghelli, Cesare Fantuzzi, Silvio Simani. (Fotocopisteria, tutorato, www)

1. Matlab, The Language of Technical Computing. Getting Started with MATLAB. *Version 5.1* (In formato pdf su CD Matlab)
2. MATLAB Primer. Second Edition. Kermit Sigmon. Department of Methematics. University of Florida.
3. The MathWorks Inc., Matlab User's Guide, 1993.
4. L. F. Shampine and M. W. Reichel, "The Matlab Ode Suite", Tech. Rep., The MathWorks, Inc, 1997. (Disponibile anche come file in formato pdf).

⇛ The MathWorks Inc., Simulink User's Guide, 1995.

5. B. C. Kuo, Automatic Control Systems. Englewood Cliffs, New Jersey: Prentice Hall, 7th ed., 1995.
6. P. Bolzern, R. Scattolini, and N. Schiavoni, Fondamenti di controlli automatici. Milano: McGraw- Hill, I ed., Marzo 1998.
7. G. Marro, TFI: insegnare e apprendere i controlli automatici di base con matlab. Bologna: Zanichelli, I ed., Ottobre 1998.
8. C. Fantuzzi, Controllori Standard PID. Versione 1.2, Appunti del Corso, 1a ed., Maggio 1997.

Integrazione in Matlab

File Ode

⇒ Un *Ode file* è un file di tipo `.m` per definire un sistema di equazioni differenziali

⇒ Un *Ode file* è risolto dagli *Ode Suite Solvers*

`y = odefile(t,x,flag,params)`

⇒ `t` e `x` sono variabili di integrazione

⇒ `flag` è una stringa che indica il tipo di informazione restituita dall' *Ode file*. `params` sono parametri addizionali eventualmente richiesti.

Integrazione in Matlab

- **Ode Suite Solvers**
- **Ode Suite Solvers: risolutori di equazioni differenziali (sono funzioni di funzione)**
- `[t,x] = odesolver('odefile',tspan,ci,options,params)`

  ⇒ ode23, ode113, ode15s, ode23s, ode23t, ode23tb

  ⇒ `options = odeset('RelTol',1e-4,'AbsTol',1e-4,'Maxstep',1e-5);`

- `[tspan,ci,options] = odefile([],[],'init')`

Elementi di grafica in Matlab

Funzioni di grafica `plot(X,Y,S)`

⇒ Grafica il vettore `Y` in funzione di `X`

⇒ `X` e `Y` sono vettori con le stesso numero di elementi

⇒ S è una stringa formata dai caratteri:

y	yellow	.	point	-	solid
m	magenta	o	circle	:	dotted
c	cyan	x	x-mark	-.	dashdot
r	red	+	plus	--	dashed
g	green	*	star		
b	blue	s	square		
w	white	d	diamond		
k	black	v	triangle (down)		

Elementi di Grafica in Matlab

`figure, plot(t,x,'-',t,y,'--')`

- **Apertura nuova finestra grafica e visualizzazione grafici sovrapposti**

`figure, plot(t,x,'-'), hold on, plot(t,y,'--')`

- **Apertura nuova finestra grafica e visualizzazione grafici sovrapposti**

  ⇒ Sono equivalenti: `hold on` mantiene il grafico corrente

Esercizi Proposti (1)

Modello matematico di Lotka-Volterra

$$\begin{cases} \dot{x}_1(t) = a_1(1 - x_1(t)/k)x_1(t) - a_2 x_1(t)x_2(t) + u(t) & \textit{Prede} \\ \dot{x}_2(t) = -a_3 x_2(t) + a_4 x_1(t)x_2(t) & \textit{Predatori} \end{cases}$$

$\Rightarrow$ $x_1(t)$ e $x_2(t)$ numero di prede di predatori

$\Rightarrow$ $u(t)$ cibo per le prede

$\Rightarrow$ k, numero massimo di prede in assenza di predatori e di cibo $(u(t) = 0)$

Esercizi Proposti (1)

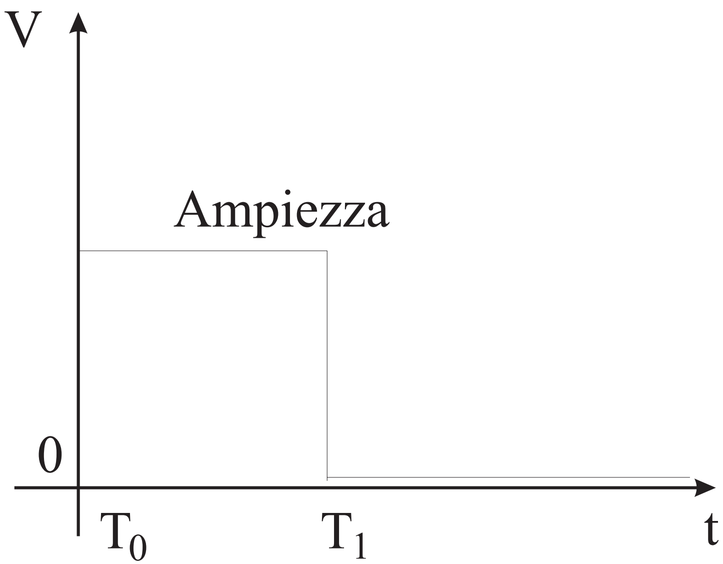
Modello matematico di Lotka-Volterra

$\Rightarrow$ Se $a_1 = 20$, $a_2 = 1$, $a_3 = 7$, $a_4 = 0.5$ e $k = 30$, si determinino:

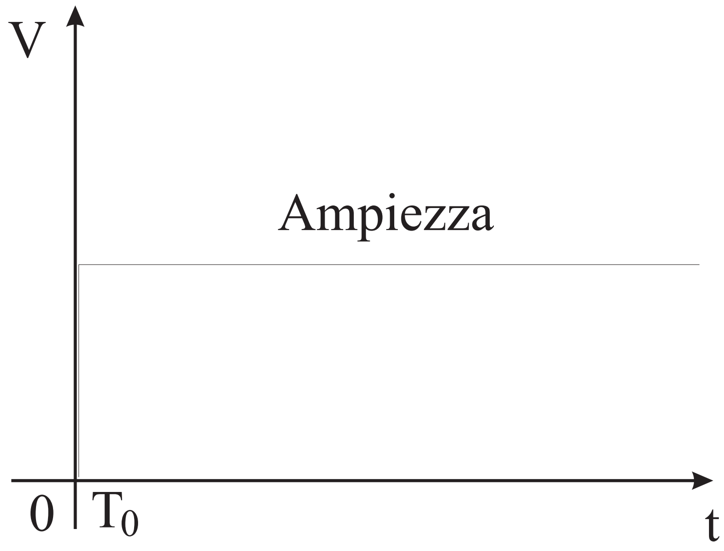
1) l'andamento nel tempo del numero di prede e predatori, supponendo nullo l'ingresso u(t) e nelle ipotesi di partire da un ecositema contenente 10 prede e 10 predatori. Si calcoli anche la traiettoria percorsa dal sistema nello spazio degli stati.
2) gli stati di equilibrio del sistema in assenza di ingresso.
3) i valori di regime raggiunti dal numero di prede e predatori nelle ipotesi che $u(t)$ sia un gradino di ampiezza $u(t) = 20$ e a partire dalle stesse condizioni proposte al punto 1). Si determini per tentativi l'ampiezza del gradino che consente di mantenere a regime un numero di predatori pari a 15.

Esercizi Proposti

Modello matematico di Lotka-Volterra



(a) Circuito diodo tunnel



(b) Modello Lotka-Volterra

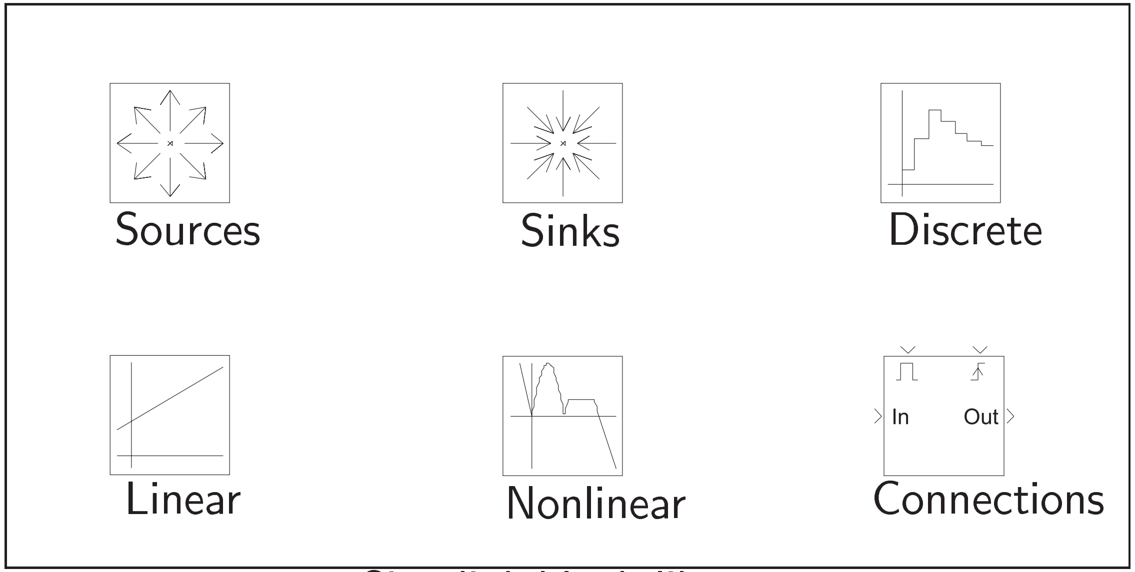
Introduzione a Simulink

- ***Simulink*, prodotto dalla *Mathworks Inc.***
- **È un programma per la simulazione di sistemi dinamici**
- **Progetto di un sistema dinamico**
  - ⇒ Definizione del modello da simulare
  - ⇒ Analisi del sistema
- **Ambiente a finestre ed interfaccia grafica: *Block diagram windows* e mouse**
- ***Simulink* riutilizza l'ambiente e i comandi di *Matlab***
  - ⇒ I risultati sono disponibili nel *Workspace* di *Matlab*

Istruzioni di base di Simulink

>> `simulink`

⇒ Visualizzazione della finestra *Simulink block library*



Simulink block library.

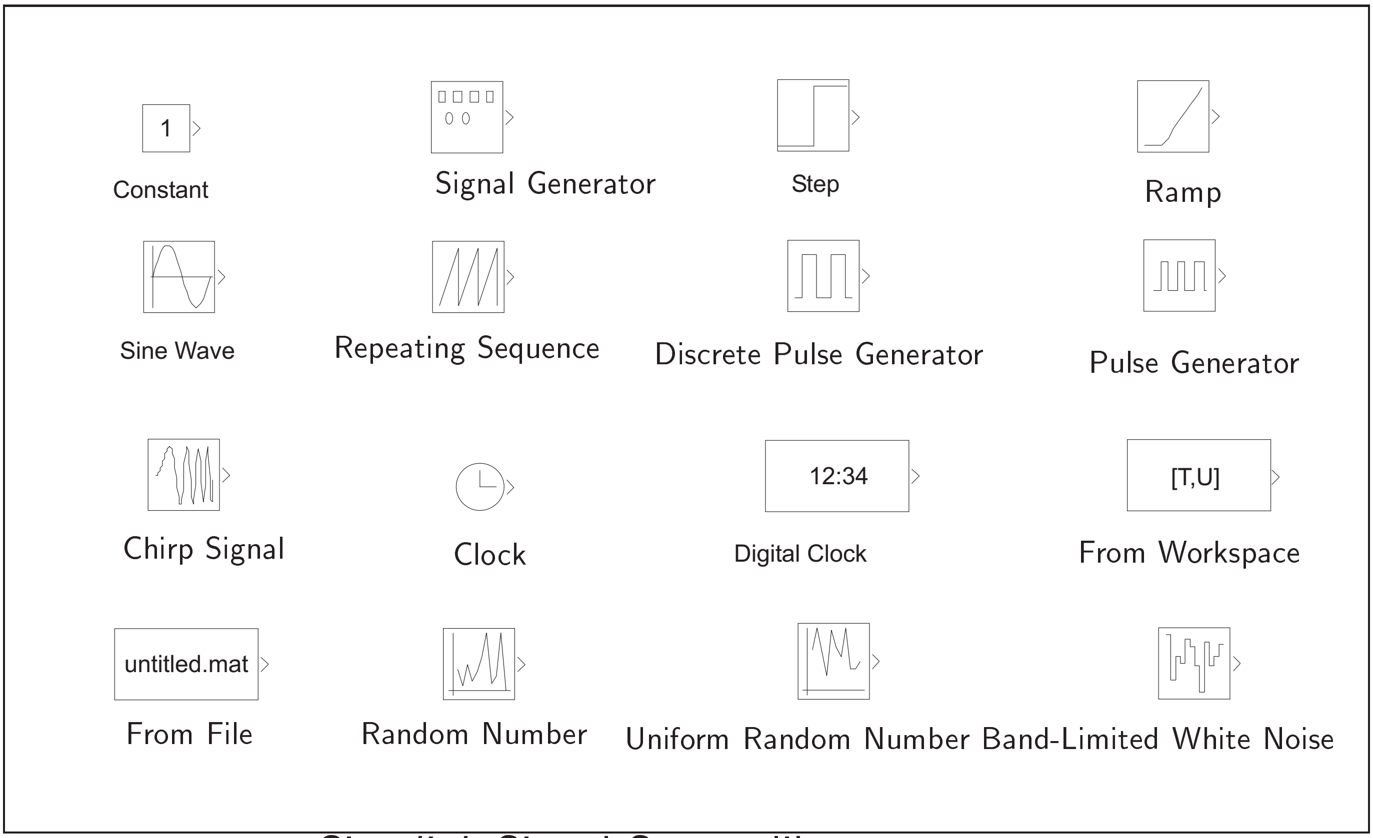
Università di Ferrara, Dip. di Ingegneria
v. Saragat, 1, I-44100, Ferrara, Italia

Silvio Simani

Istruzioni di base di Simulink

Sources Library

⇒ **Sources**: (Library: simulink/Sources), generatori di segnale

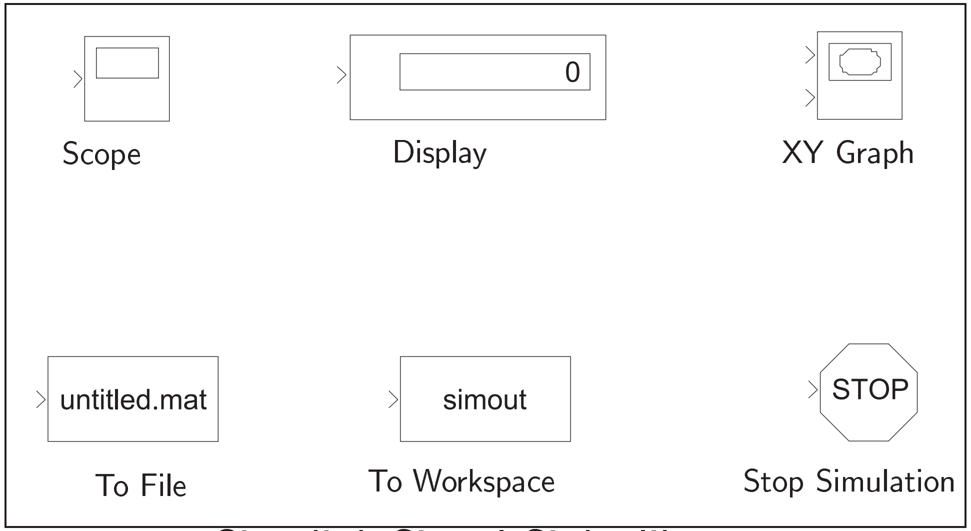


Simulink Signal Source library.

Istruzioni di base di Simulink

Sink Library

⇒ *Library: simulink/Sinks* contiene alcuni rivelatori di segnale

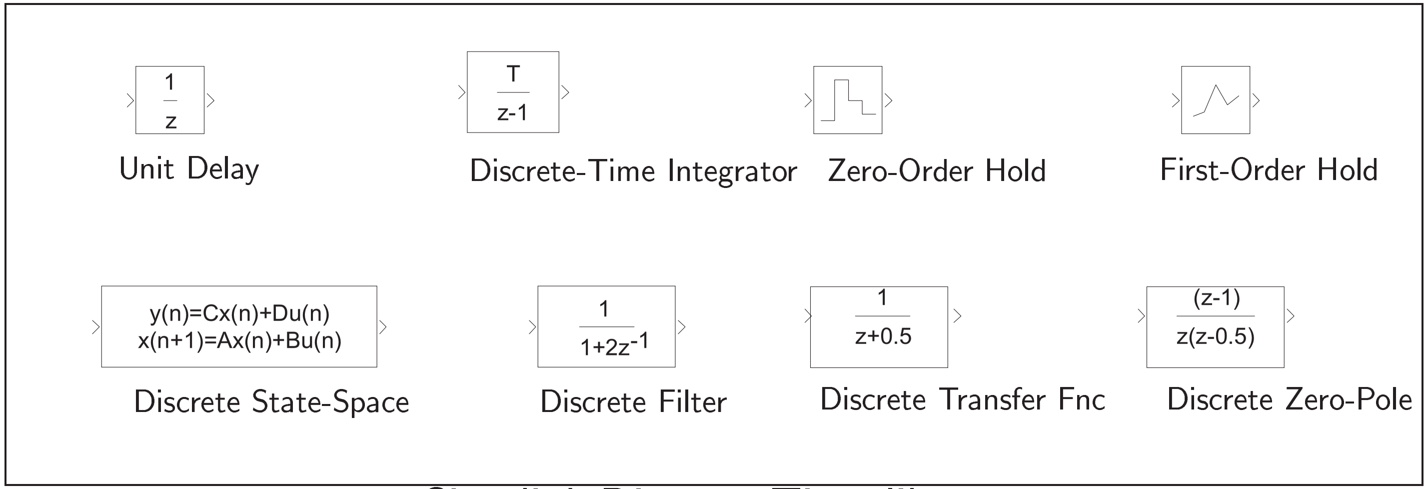


Simulink Signal Sinks library.

Istruzioni di Simulink

Discrete Library

⇒ *Library: simulink/Discrete* analisi dei sistemi lineari tempo-discreti

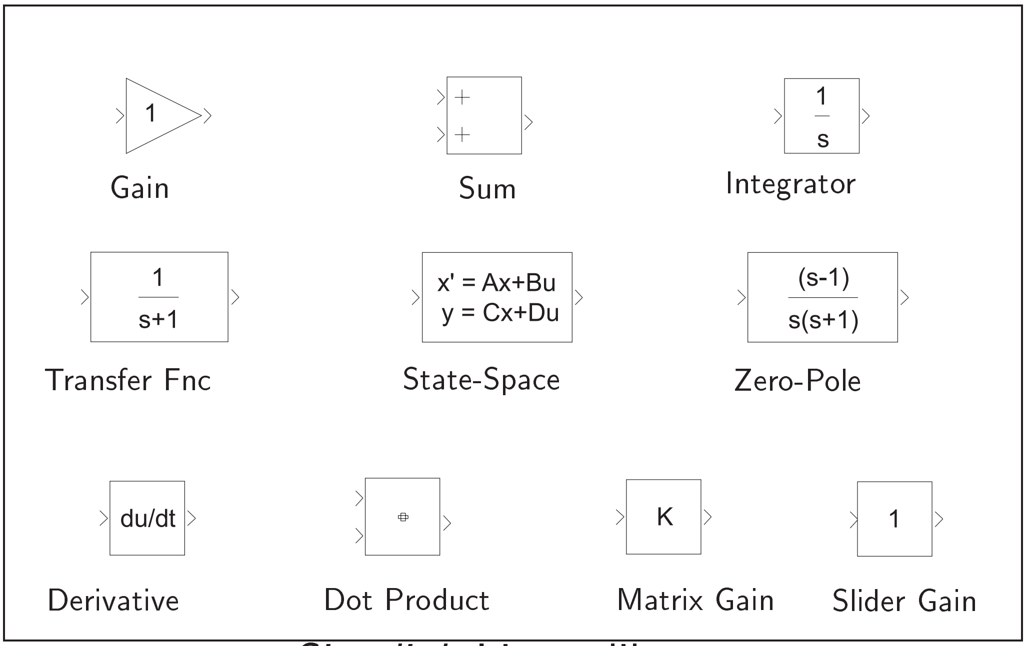


Simulink Discrete-Time library.

Istruzioni di Simulink

Linear Library

⇒ *Library: simulink/Linear* analisi dei sistemi lineari tempo-continui

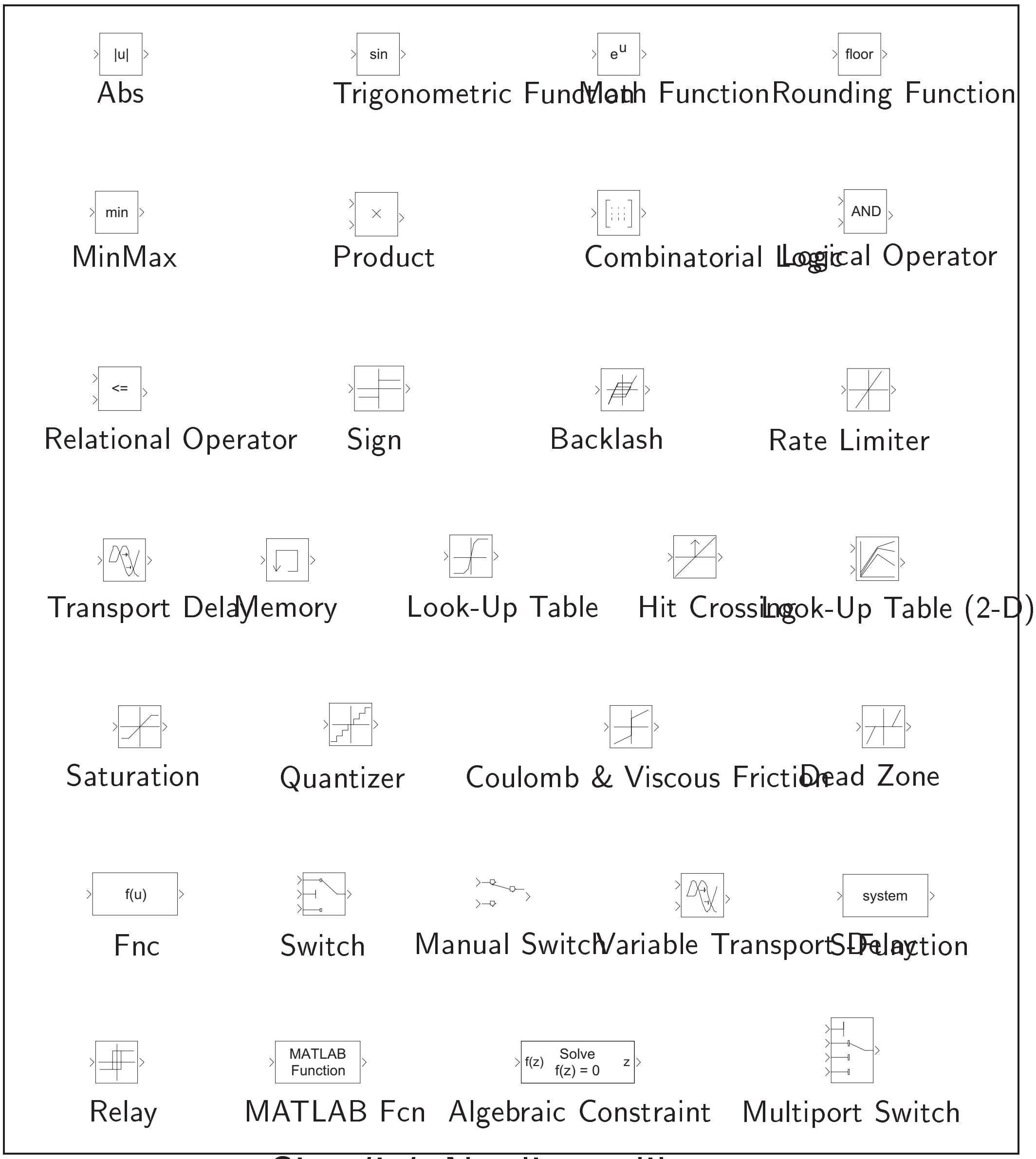


Simulink Linear library.

Istruzioni di Simulink

Nonlinear Library

⇒ *Library: simulink/Nonlinear* funzioni non lineari

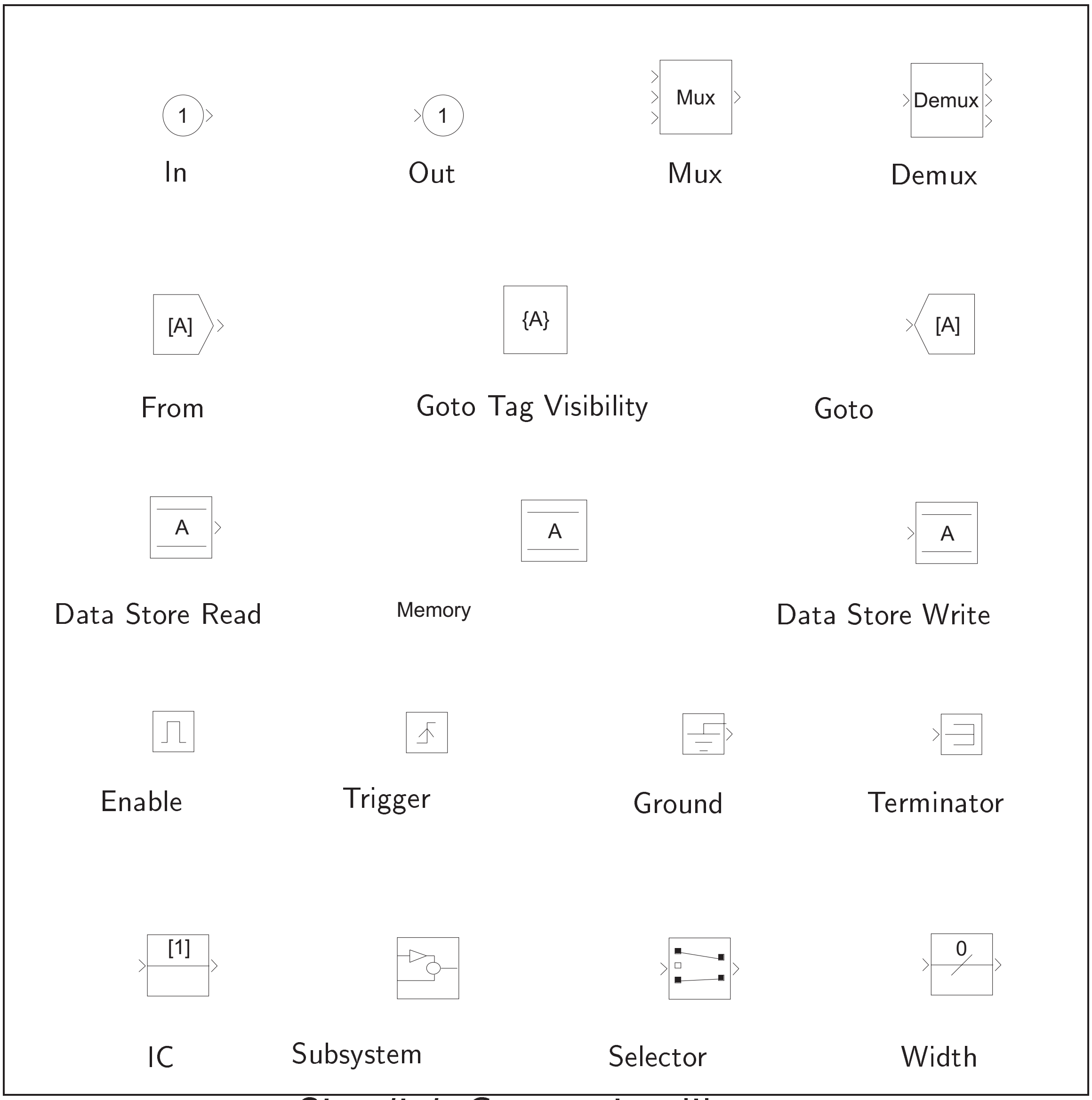


Simulink Nonlinear library.

Istruzioni di Simulink

Connections Library

⇒ *Library: simulink/Connections* blocchi per effettuare connessioni



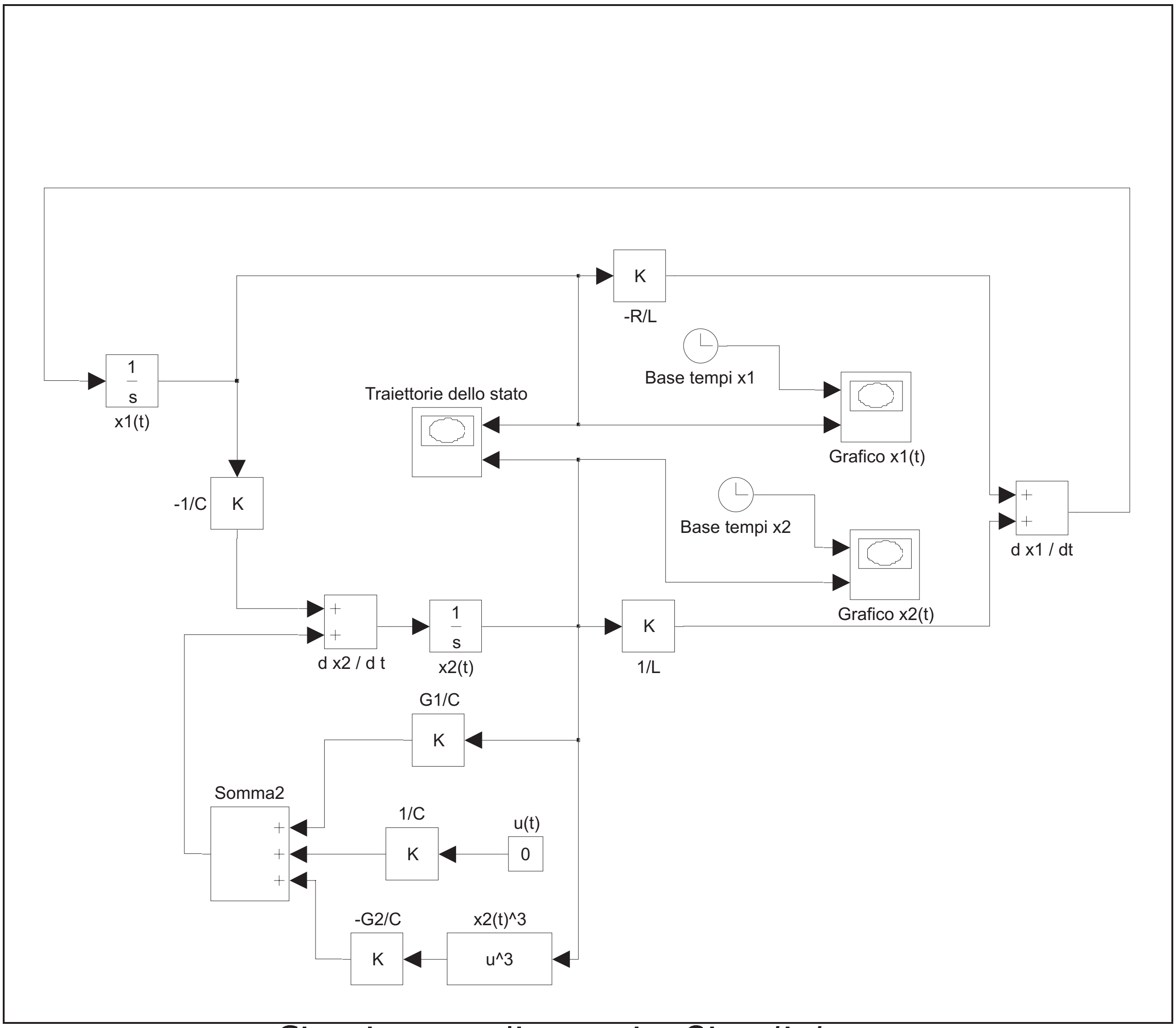
Simulink Connection library.

Simulazione di sistemi dinamici con SIMULINK

- **Simulazione $\hookrightarrow$ integrazione delle equazioni differenziali**
- **Simulatore di *Simulink***
- ***Simulink Control Panel* $\hookrightarrow$ *Simulation* $\hookrightarrow$ *Parameters***
- **Opzione *Solver* nella finestra *Simulation Parameters* (cfr. odesolver *Matlab*)**
  - $\Rightarrow$ **Simulation time: Start time**: istante iniziale della simulazione
  - $\Rightarrow$ **Simulation time: Stop time**: istante finale della simulazione
  - $\Rightarrow$ **Solver Options: Type**: passo di integrazione fisso (Fixed-step) o variabile (Variable-step)
  - $\Rightarrow$ **Solver Options**: scelta della funzione di integrazione ottimale: `ode45`, `ode23`, `ode113`, `ode15s`, `ode23s`, e `discrete`
  - $\Rightarrow$ **Solver Options**: Max step size e Initial step size, Relative e Absolute tolerance

Analisi di un circuito non lineare

$$
\begin{aligned}
\dot{x}_1(t) &= -\frac{R}{L}\, x_1(t) + \frac{1}{L}\, x_2(t) \\
\dot{x}_2(t) &= -\frac{1}{C}\, x_1(t) + \frac{1}{C}\big(G_1 x_2(t) - G_2 x_2^3(t)\big) + \frac{1}{C} u(t)
\end{aligned}
$$



Circuito non lineare in *Simulink*.

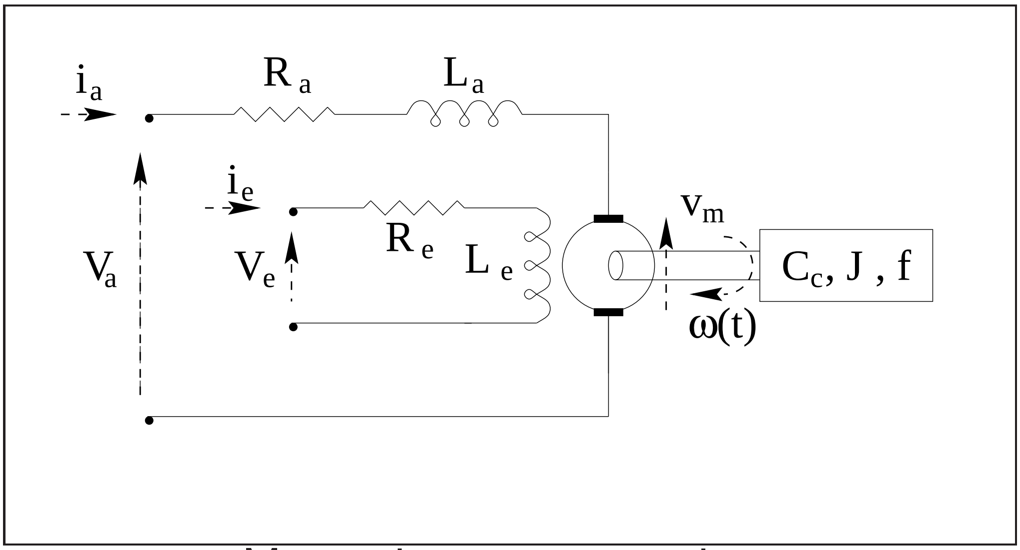
Modello di un motore in corrente continua

Controllo d'armatura e avvolgimento d'eccitazione ad alimentazione costante

$$\begin{bmatrix} \dot{i}_a(t) \\ \dot{\omega}(t) \end{bmatrix} = \begin{bmatrix} -\frac{R_a}{L_a} & -\frac{k_m}{L_a} \\ \frac{k_m}{J} & -\frac{f}{J} \end{bmatrix} \begin{bmatrix} i_a(t) \\ \omega(t) \end{bmatrix} + \begin{bmatrix} \frac{1}{L_a} & 0 \\ 0 & -\frac{1}{J} \end{bmatrix} \begin{bmatrix} V_a(t) \\ C_c(t) \end{bmatrix}$$

$$\omega(t) = \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} i_a(t) \\ \omega(t) \end{bmatrix}$$

Motore in corrente continua



Motore in corrente continua

Modello di un motore in corrente continua

- **Matrici di sistema**

$$A = \begin{bmatrix} -\frac{R_a}{L_a} & -\frac{k_m}{L_a} \\ \frac{k_m}{J} & -\frac{f}{J} \end{bmatrix}, \; B = \begin{bmatrix} \frac{1}{L_a} & 0 \\ 0 & -\frac{1}{J} \end{bmatrix} \text{ e } C = \begin{bmatrix} 0 & 1 \end{bmatrix}.$$

- **Ingressi ed uscite:** $y(t) = \omega(t)$

$$x(t) = \begin{bmatrix} i_a(t) \\ \omega(t) \end{bmatrix} \text{ e } u(t) = \begin{bmatrix} V_a(t) \\ C_c(t) \end{bmatrix}.$$

Modello di un motore in corrente continua

Condizioni di funzionamento

1. Alimentazione del motore con un gradino di tensione di armatura costante a 0V da 0s a 50s e al valore di 5V per altri 50s.
2. Impulso di tensione di ampiezza $V_a(t) = 10$V e durata $\tau = 40$s.
3. Posizione del rotore $\alpha(t)$

$$\dot{\alpha}(t) = \omega(t)\ ,\ x(t) = \begin{bmatrix} i_a(t) \\ \omega(t) \\ \alpha(t) \end{bmatrix} \text{ e } y(t) = \begin{bmatrix} \omega(t) \\ \alpha(t) \end{bmatrix}$$

$$A = \begin{bmatrix} -\frac{R_a}{L_a} & -\frac{k_m}{L_a} & 0 \\ \frac{k_m}{J} & -\frac{f}{J} & 0 \\ 0 & 1 & 0 \end{bmatrix}\ ,\ B = \begin{bmatrix} \frac{1}{L_a} & 0 \\ 0 & -\frac{1}{J} \\ 0 & 0 \end{bmatrix} \text{ e } C = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

Esercizi proposti

1. Si realizzi in ambiente *Simulink* il sistema di equazioni differenziali relative ai modello del motore in corrente continua. Se ne verifichi successivamente la correttezza confrontandolo con le realizzazioni equivalenti nello spazio degli stati.
2. Utilizzando gli stessi i valori dei parametri del motore in corrente continua, determinare l'ampiezza del gradino $V_a(t)$ necessaria a raggiungere una velocità angolare *di regime* pari a $\omega(t) = 10$rad/s, nelle ipotesi di assenza del carico $C_c = 0$ e con il modello del motore del secondo ordine. Si verifichi analiticamente il risultato ottenuto.
3. Fissata l'ampiezza della tensione di armatura a $V_a = 10$V., progettare la durata dell'impulso τ in modo da raggiungere una posizione assegnata $\alpha = 20$rad., sempre nelle ipotesi di assenza di carico.
4. Fissato τ, graficare l'andamento temporale della posizione del rotore per una tensione pari alla metà e al doppio della tensione fissata al punto precedente.