Principi di Bioingegneria
A.A. 2024/25
Lezione

Simulazioni numeriche

Vincenzo Catrambone, PhD

vincenzo.catrambone@unipi.it

Integrazione numerica

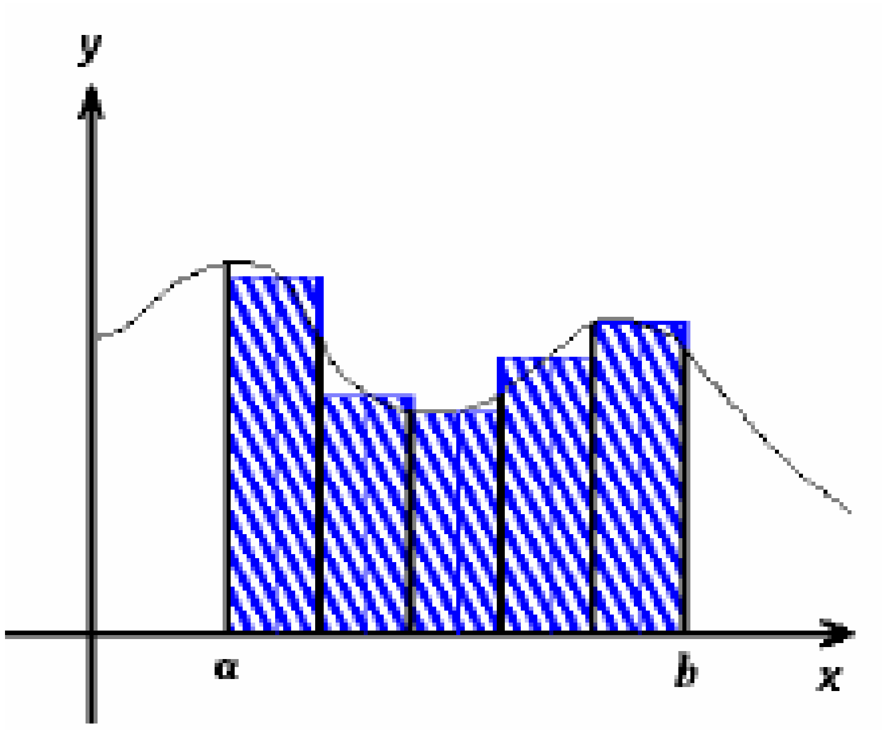
Derivazione numerica

Predizione numerica

ODE in MATLAB

Simulink

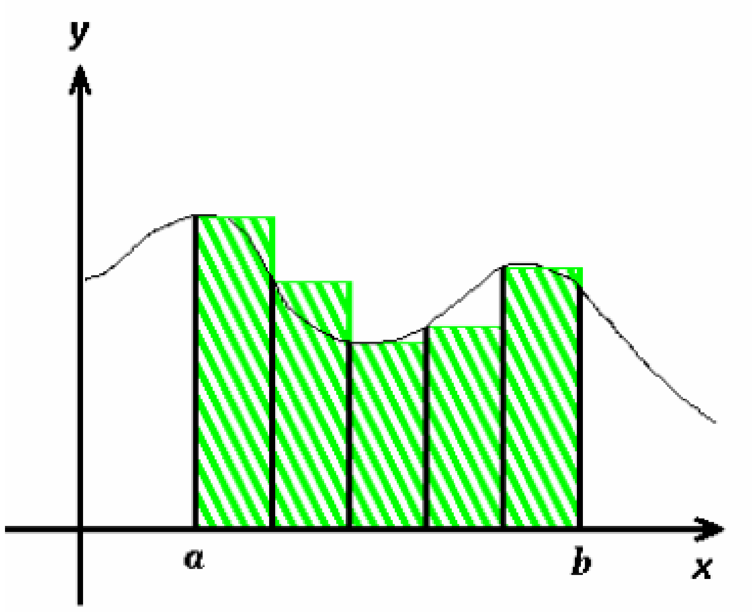
Integrazione numerica

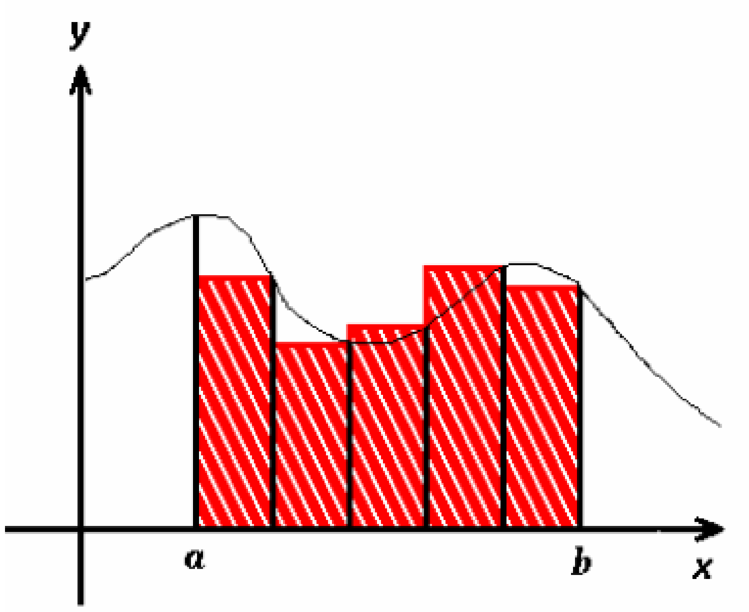


Metodo dei rettangoli:
Si calcola l'integrale di una funzione come sommatoria delle aree dei rettangoli individuati da intervalli definiti sul dominio della funzione e altezza pari al valore ottenuto dalla funzione nel punto medio dell'intervallo

Integrazione numerica

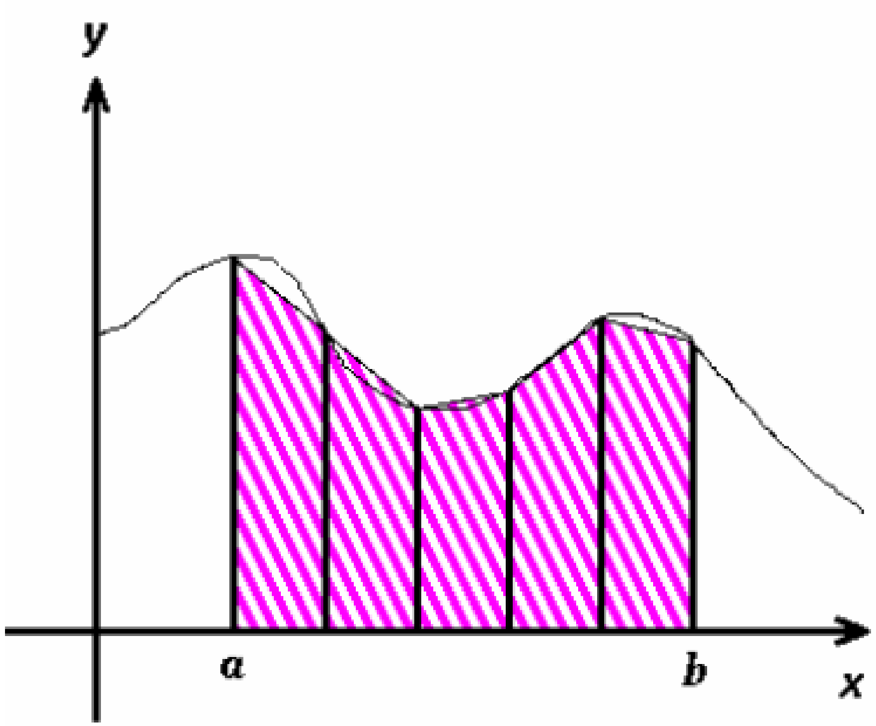
Altre due varianti: altezza su punto sinistro, altezza su punto destro





Integrazione numerica

Metodo dei trapezi:
Le aree delle sezioni vengono calcolate come aree di trapezi rettangoli



Funzioni predefinite in MATLAB

trapz(x,y): applica il metodo dei trapezi per calcolare l'integrale di y(x). y deve contenere i valori della funzione in corrispondenza dei punti contenuti in x.

Es.

```
>> x = 0:0.1:pi;
>> y = sin(x);
>> trapz(x,y)
```

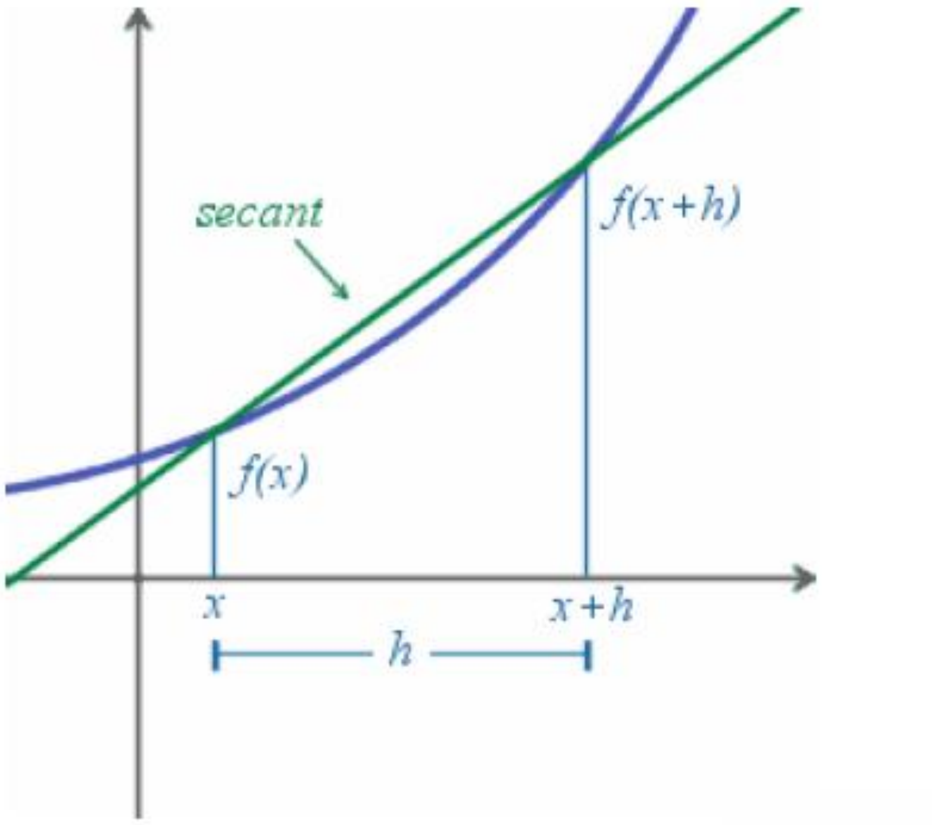
quad('fun',a,b,tol) : Applica la regola di Simpson per calcolare l'integrale di *'fun'* tra *a* e *b*. *tol* (tolleranza di errore) è opzionale

$$\mathcal{I}_{CS}(f) = \frac{(b-a)}{6}\left(f(a) + 4f\left(\frac{a+b}{2}\right) + f(b)\right).$$

quadl('fun',a,b,tol) : Applica il metodo di quadratura di Gauss-Legendre-Lobatto. Sintassi analoga a quad.

$$\mathcal{I}_G^c(f) = \frac{H}{2}\sum_{k=1}^{N}\sum_{i=0}^{n} w_i f(x_{ki}) \quad H = \frac{b-a}{N}$$

Derivazione numerica

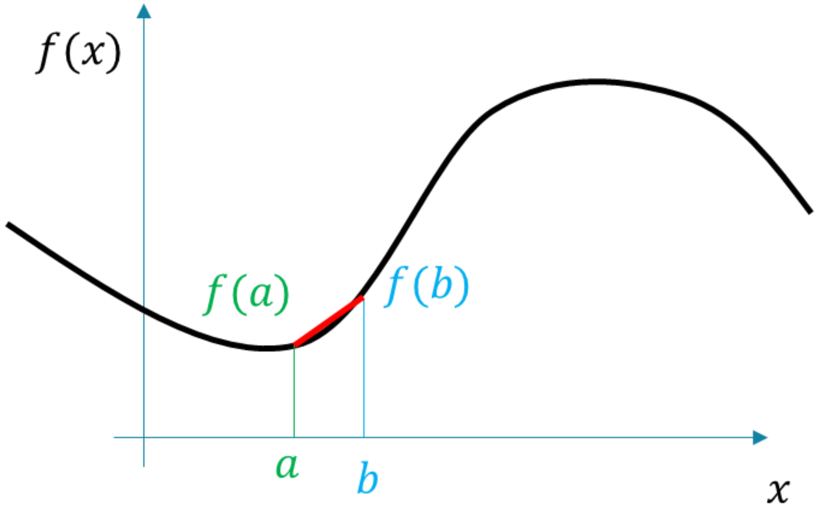


$$f'(x) = \lim_{h \to 0} \frac{f(x+h) - f(x)}{h}$$

Derivazione numerica

Differenze in avanti

$$f'(x) = \frac{f(x) - f(x+\delta)}{(x) - (x+\delta)}$$



$$f'(a) = \frac{f(a) - f(b)}{a - b}$$

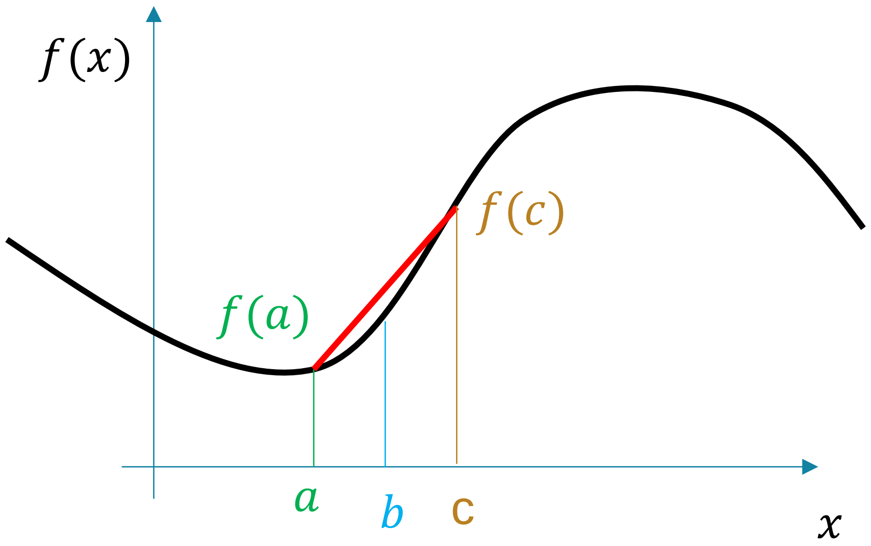
Differenze in indietro

$$f'(x) = \frac{f(x+\delta) - f(x)}{(x+\delta) - (x)}$$

$$f'(b) = \frac{f(b) - f(a)}{b - a}$$

Differenze centrali

$$f'(x) = \frac{f(x+\delta) - f(x-\delta)}{(x+\delta) - (x-\delta)}$$

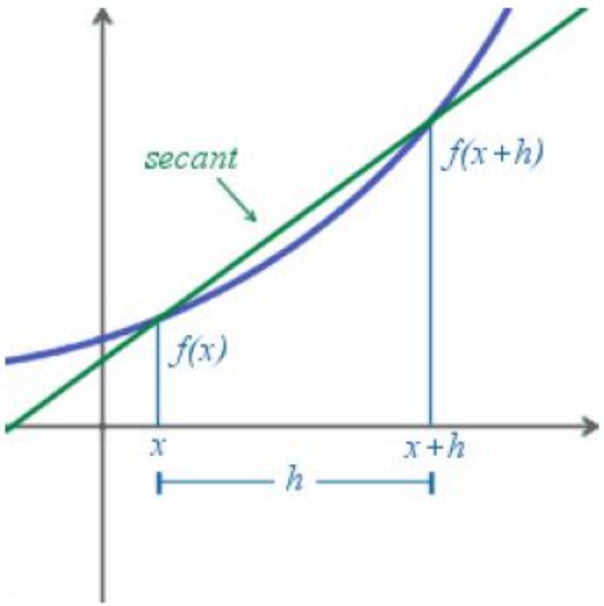


$$f'(b) = \frac{f(c) - f(a)}{c - a}$$

Metodi di predizione: metodo di Eulero

Il metodo delle differenze, e relative varianti portano poi alla definizione del metodo di Eulero, che sfrutta la stima della derivata di una funzione per predire i valori che la funzione assume in zone del dominio non (ancora) campionate.

$$\frac{dy}{dt} = f(y,t)$$



$f(y,t)$ può essere stimata con qualunque dei metodi espressi in precedenza e si applica in modo da ottenere

$$\widehat{y_{n+1}} = y_n + hf(y_n, t)$$

poi in maniera ricorsiva

$$\widehat{y_1} = y_0 + hf(y_0, t_0)$$
$$\widehat{y_2} = y_1 + hf(y_1, t_1)$$

$$\widehat{y_n} = y_{n-1} + hf(y_{n-1}, t_{n-1})$$

$\widehat{y_i}$ è il valore stimato predetto
y_i è il valore reale

Dove h è lo step size ed è legato a $t_{n+1} - t_n$: può essere ridotto in modo da avere una griglia temporale più fine ma incide sulla precisione della soluzione

Metodo di Eulero Modificato (o del punto medio esplicito)

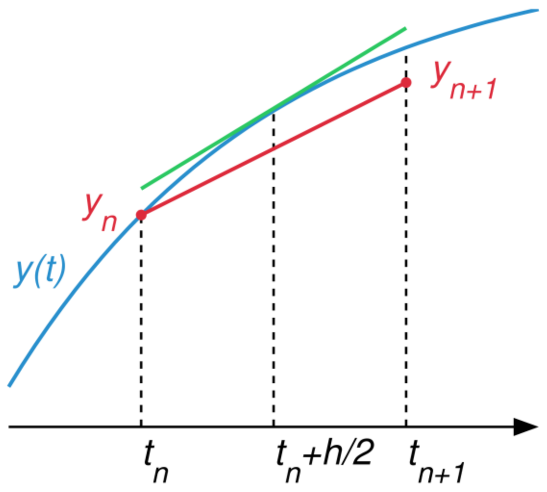
Più accurato e stabile di quello di Eulero in avanti

Si usa la conoscenza della derivate nel punto medio tra t_n e t_{n+1} (e non quella in t_n) per calcolare il valore di y_{n+1}

Per conoscere il valore di y nel punto medio tra t_n e t_{n+1} possiamo usare Eulero e usare la derivate nel punto iniziale

$$y_{n+1} = y_n + hf\left(y_n + \frac{h}{2}\mathrm{f}(y_n, t_n), t_n + \frac{h}{2}\right)$$

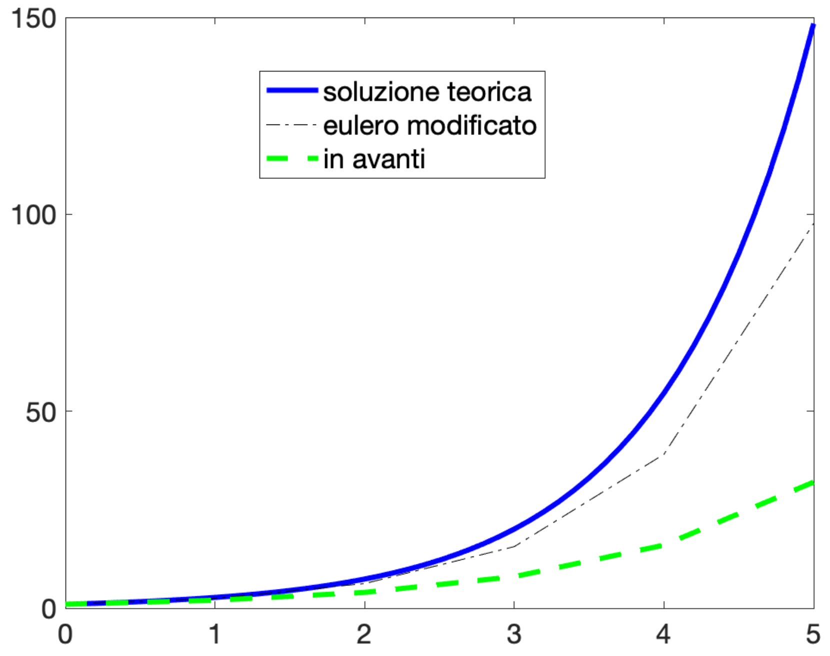
si potrebbe anche usare una stima della derivata nel punto medio (metodo implicito)



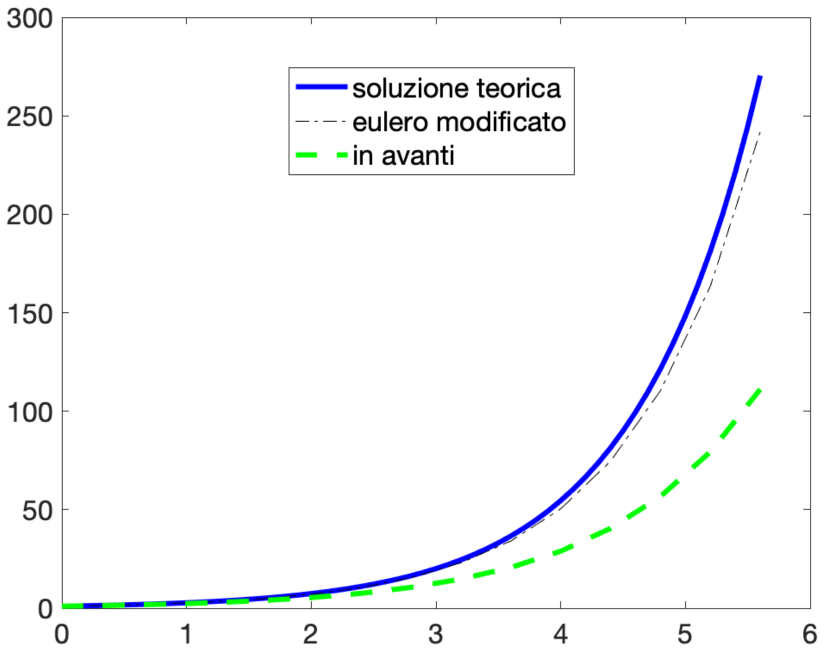
Metodo di Eulero

Soluzione teorica in blu dell’equazione

in verde il metodo di Eulero in avanti e in nero tratteggiato il metodo di Eulero modificato o del punto medio esplicito



Passo h = 1



Passo h = 0.4

Metodo Runge-Kutta

In questo caso il punto al tempo n+1 viene stimato utilizzando le stime delle derivate nel punto iniziale, due stime differenti della derivata nel punto medio e quella nel punto finale, pesate in modo diverso. E' un metodo basato sullo sviluppo in serie di Taylor della funzione f(y(t),t)

$$\frac{dy_n}{dt} = \mathrm{f}(y_n, t_n) \quad \text{dove } y_n = y(t_n)$$

$$k_1 = \mathrm{f}(y_n, t_n)$$

$$k_2 = \mathrm{f}\left(y_n + \frac{k_1}{2}, t_n + \frac{h}{2}\right)$$

$$k_3 = \mathrm{f}\left(y_n + \frac{k_2}{2}, t_n + \frac{h}{2}\right)$$

$$k_4 = \mathrm{f}(y_n + k_3, t_n + h)$$

$$y_{n+1} = y_n + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4)$$

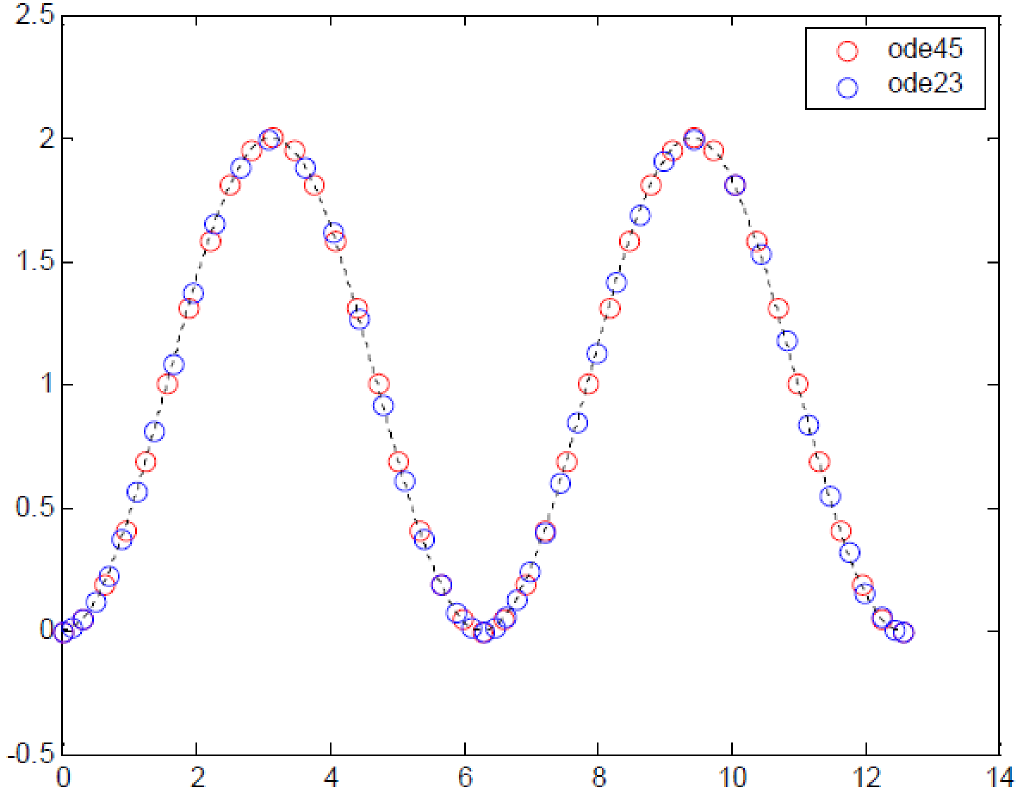
ODE: Ordinary Differential Equation

Solver	Accuracy	Description
`ode45`	Medium	This should be the first solver you try
`ode23`	Low	Less accurate than `ode45`
`ode113`	Low to high	For computationally intensive problems
`ode15s`	Low to medium	Use if `ode45` failed

ODE in MATLAB

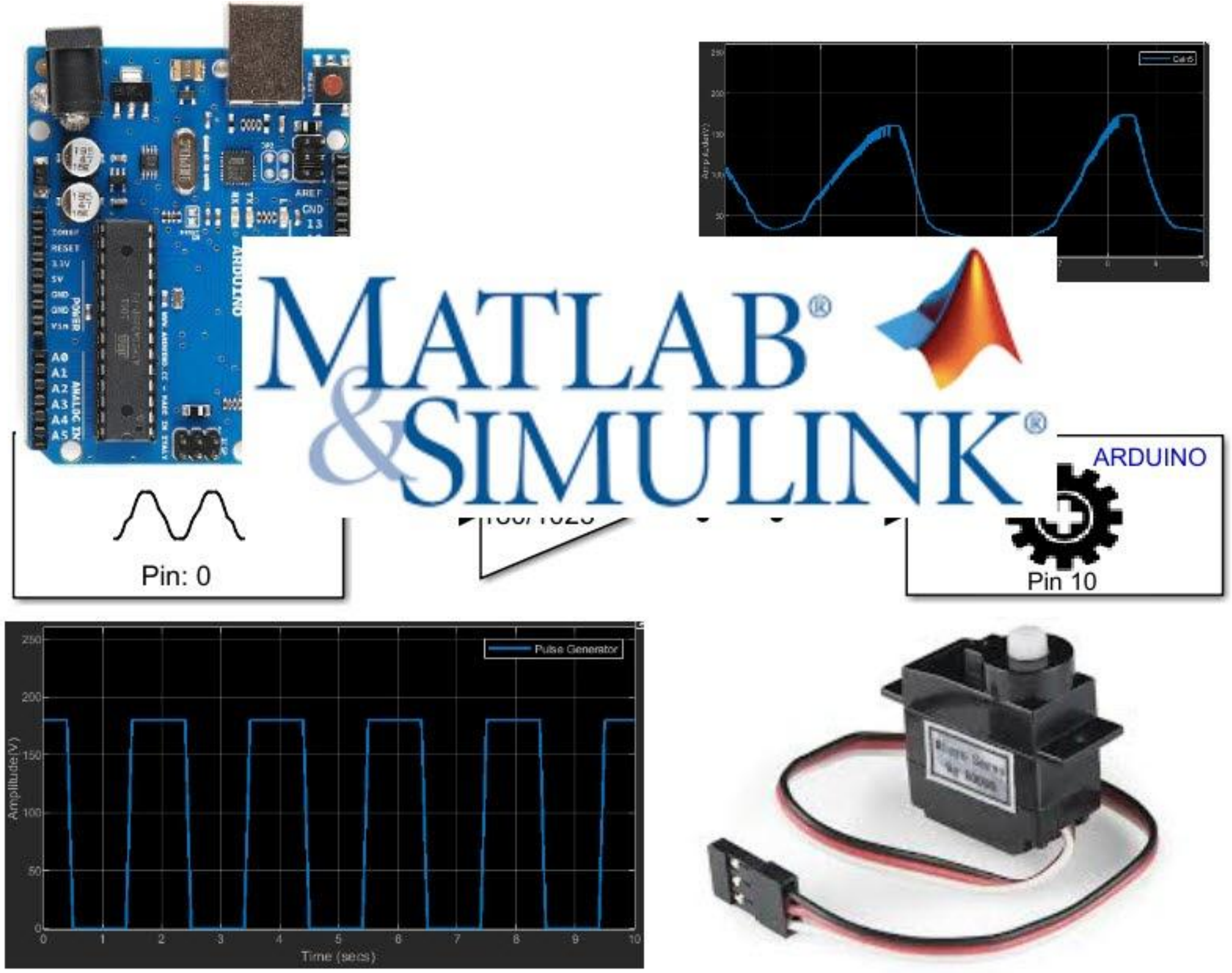
ODE: Ordinary Differential Equation

Soluzione di y' = sin(t) , con y(0) = 0




```
[t,y] = ode23('ydot', tspan, y0)
```

Risolve l'equazione differenziale y'=f(t,y) specificata nel file di funzione ydot, i cui input sono t ed y e il cui output è un vettore colonna che rappresenta dy/dt , cioè f(t,y). Il vettore tspan contiene i valori iniziale e finale della variabile t. y0 rappresenta il valore iniziale y(t0). Analogo per le altre funzioni odeXY

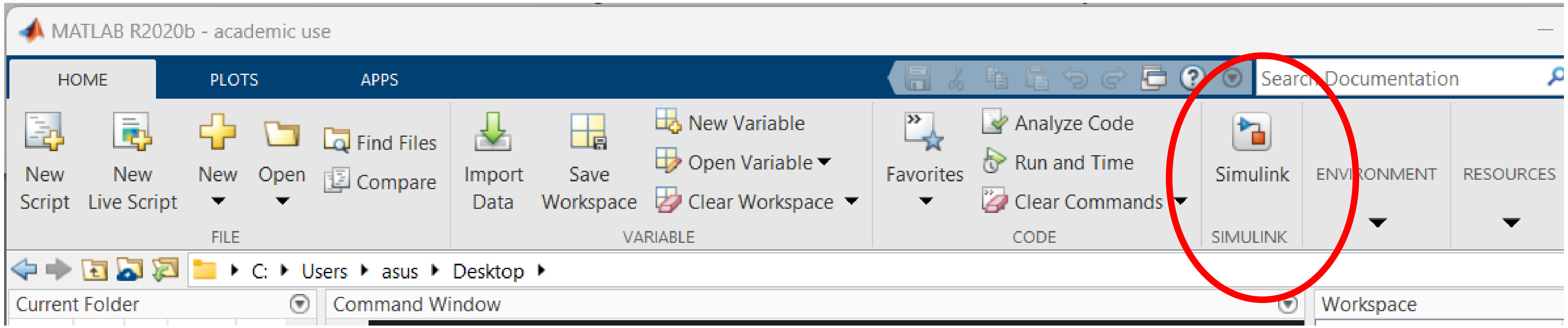

MATLAB®
&SIMULINK®
ARDUINO
Pin: 0
Pin 10
Pulse Generator
Amplitude (V)
Time (secs)

Simulink

Simulink è un programma costruito utilizzano i comandi di MATLAB, consente la realizzazione di modelli per la simulazione dinamica per via grafica

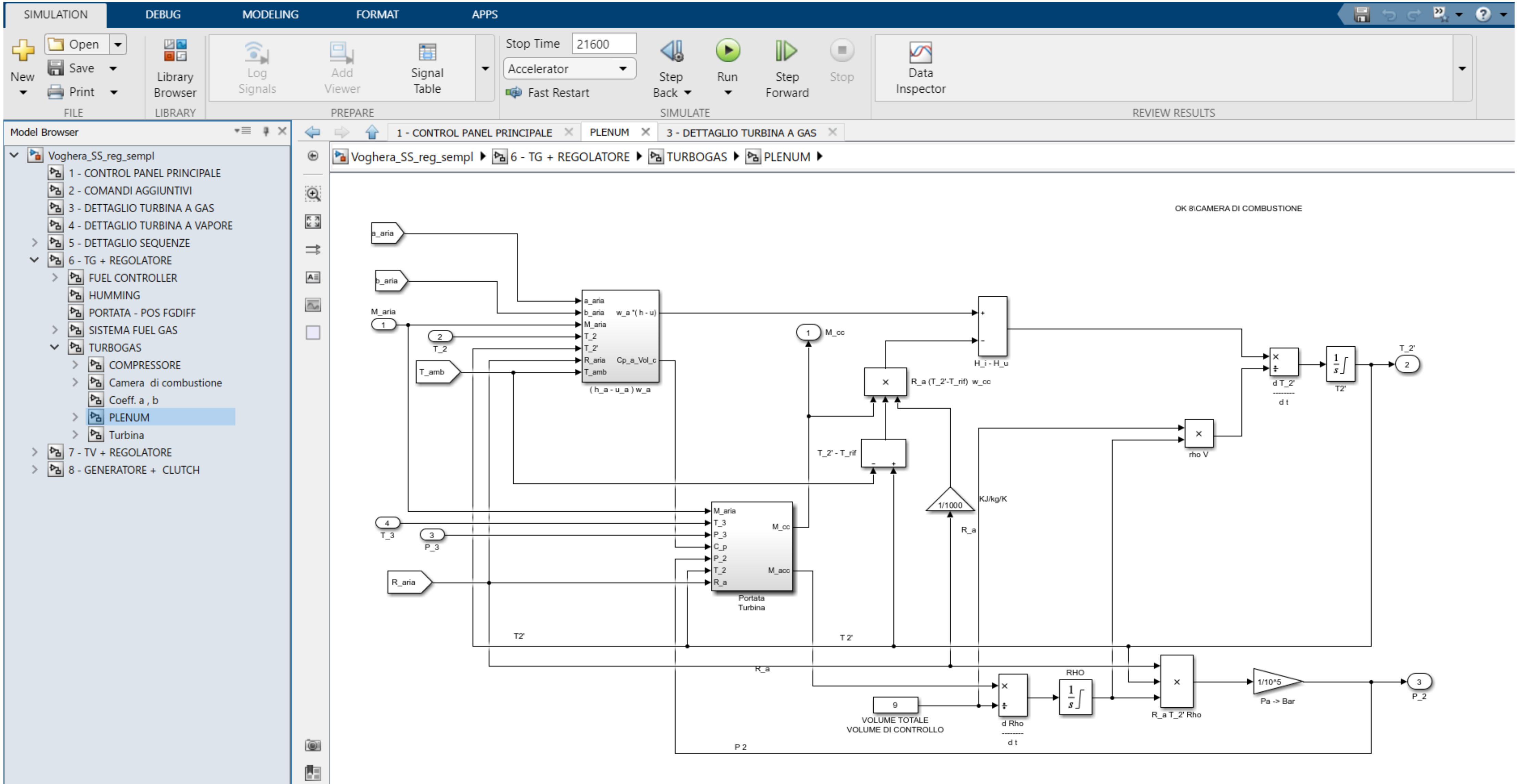
Vantaggi:
- Interfaccia grafica
- Blocchi predefiniti solamente da connettere
- Elevata flessibilità nella variazione del progetto
- Riduzione dei tempi di progetto
- Riduzione dei costi rispetto a un test pratico

Per accedere a Simulink basta digitare *simulink* dal prompt di MATLAB, o fare click sull'icona

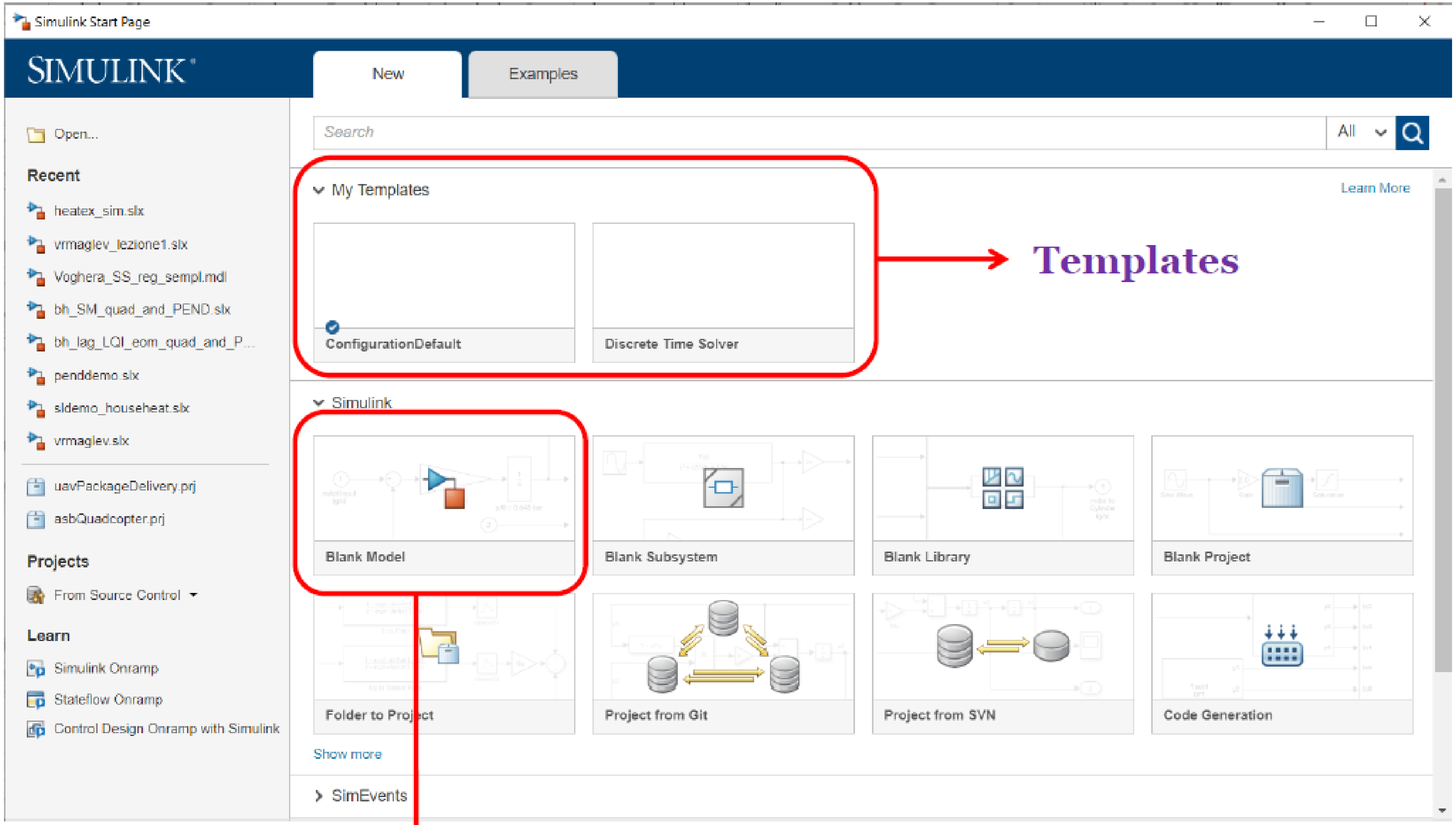


Simulink

Ciascuna linea di connessione corrisponde ad un segnale. I vari blocchi applicano ai segnali in ingresso determinate operazioni matematiche, e producono in uscita segnali risultanti da tali elaborazioni.



Simulink



Templates

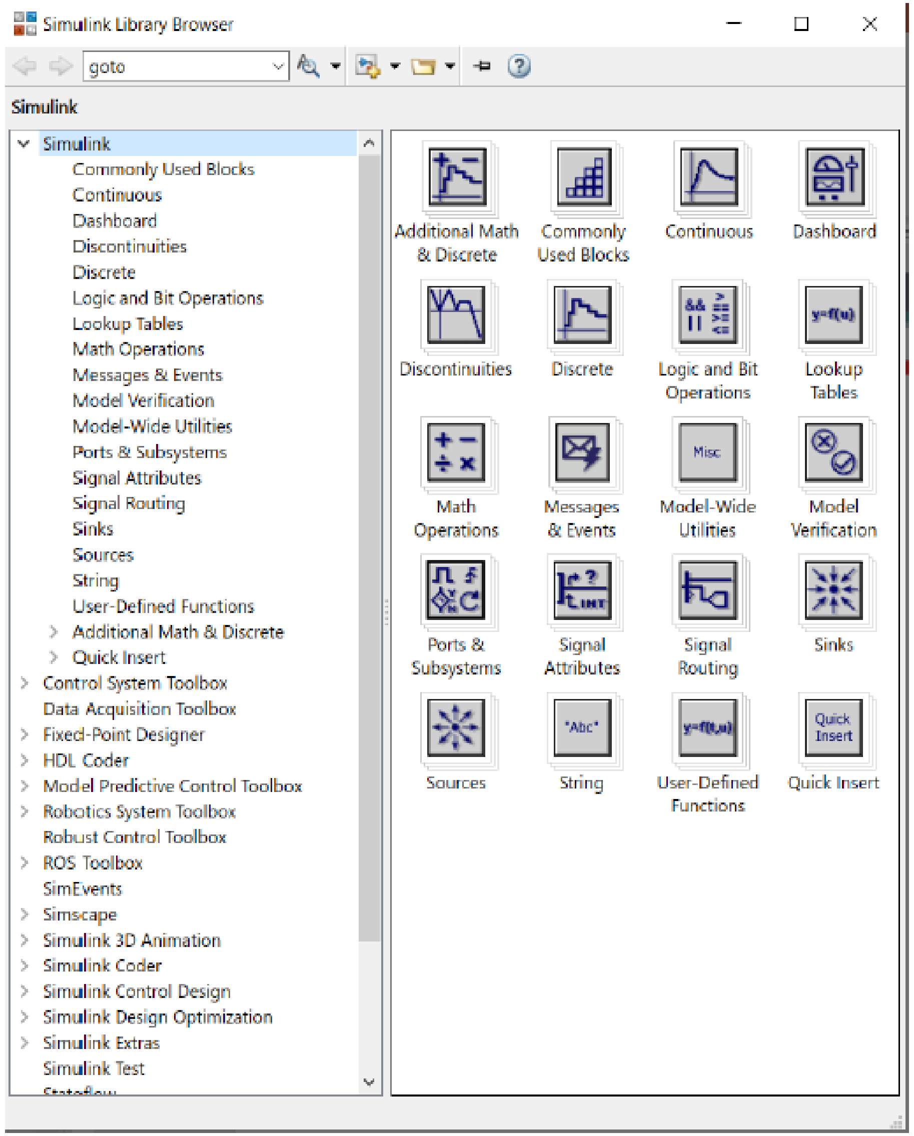
Apre un modello Simulink vuoto avente le proprietà di default

Simulink

La realizzazione di modelli di simulazione dinamica avviene per via grafica assemblando fra loro all'interno della pagina di lavoro un certo numero di blocchi Simulink, in modo da implementare le funzionalità desiderate.
I blocchi Simulink sono allocati all'interno di librerie. E' possibile accedere al «Library browser» cliccando il relativo pulsante nella finestra che ospita il modello in bianco (Menu: «SIMULATION»)

Nelle librerie sono presenti i blocchi elementari che si possono usare nel progetto.
- Nel workspace si costruisce il progetto interconnettendo i blocchi presi dalla librerie.
- I vari elementi si portano nel workspace semplicemente trascinandoli dentro come se fossero icone.
- Le librerie sono Read-only. Per poter variare i parametri di un blocco occorre prima trascinarlo nel workspace.
- Facendo doppio click sull'icona trascinata nel workspace si apre una maschera che ci consente di impostare i parametri che caratterizzano il segnale

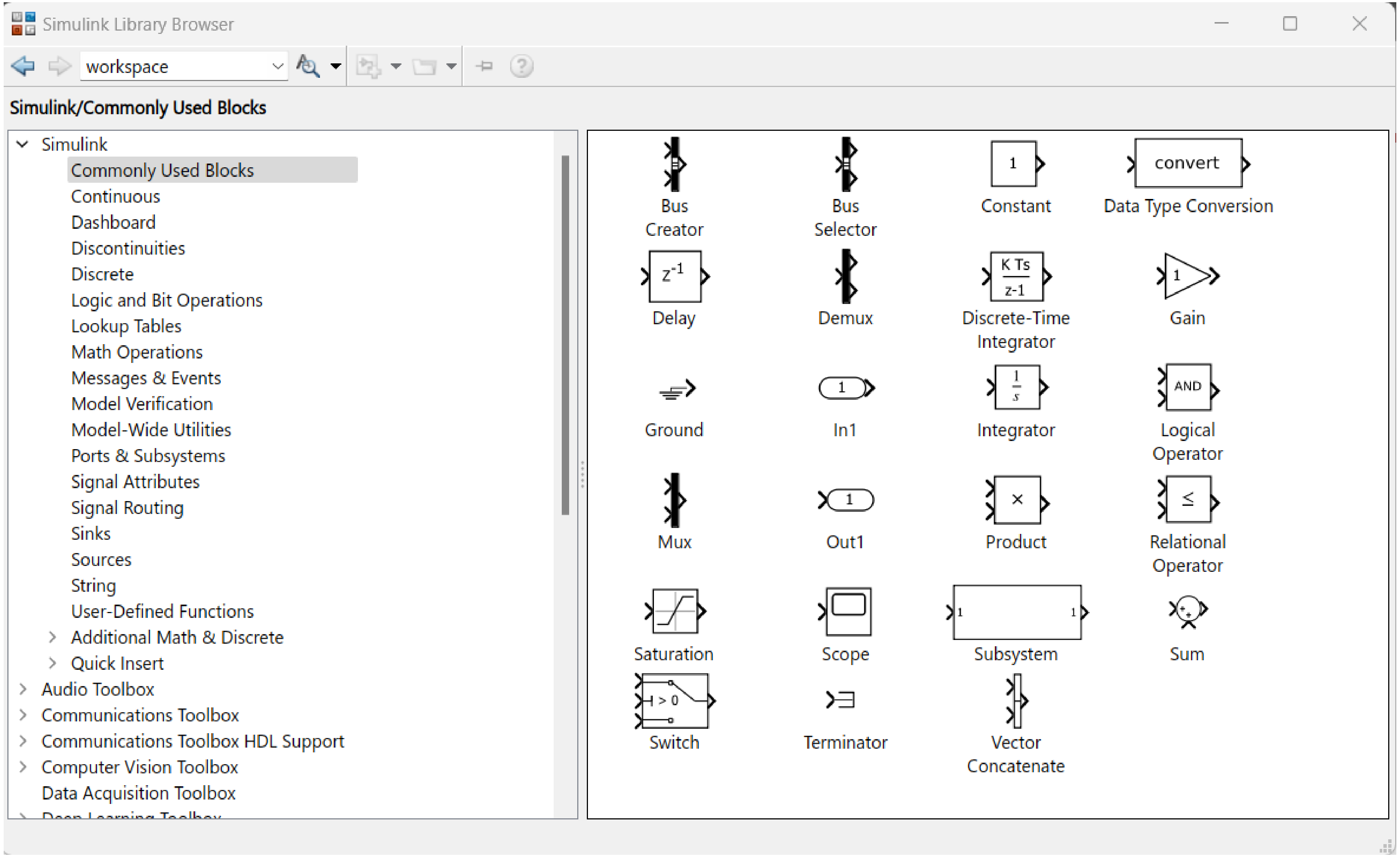
Simulink Library Browser



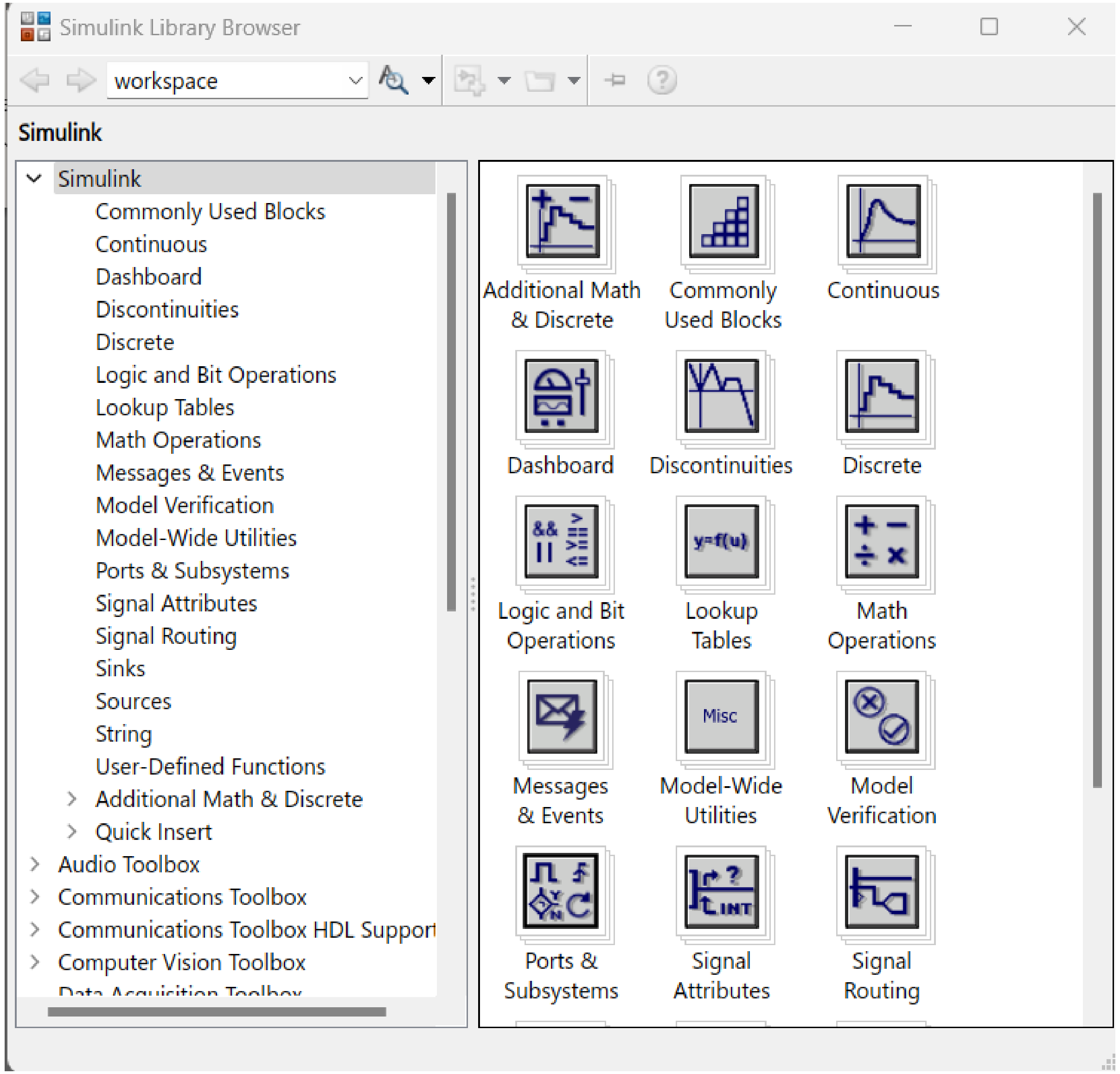
Contiene le librerie di blocchi elementari SIMULINK, da quelli di base fino a quelli più sofisticati orientati a particolari applicazioni

Simulink: libraries

La prima libreria della lista è la Libreria "**Commonly used blocks**"
E' una libreria che contiene un insieme di blocchi che implementano funzionalità variegate e che vengono impiegati particolarmente di frequente nei modelli di simulazione.



Simulink: libraries



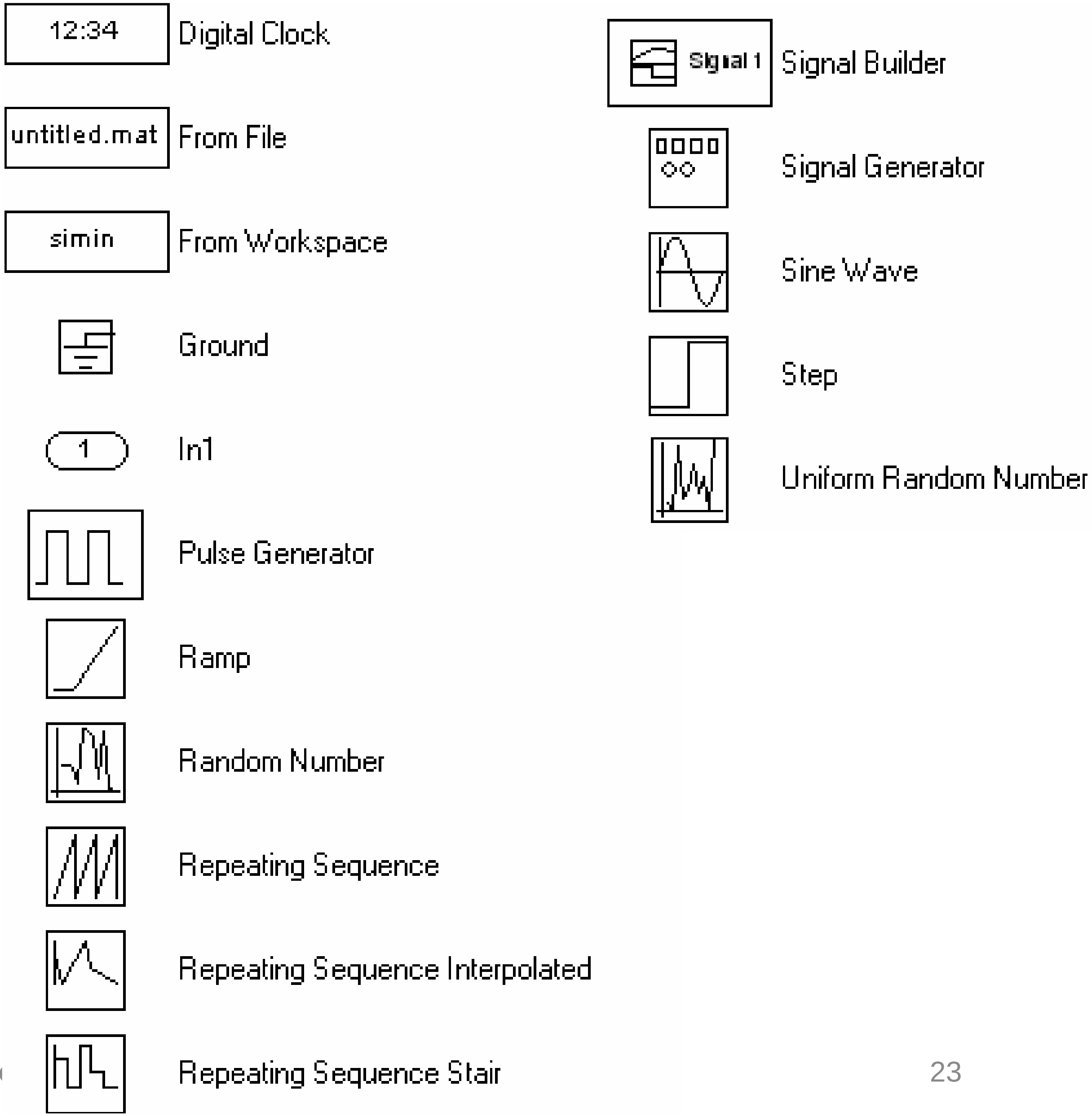
Fa le svariate librerie, le più utilizzate sono:

- Sources: Blocchi che generano segnali di vario genere
- Sinks: Blocchi per la visualizzazione grafica dei segnali
- Math: Blocchi per l'elaborazione matematica dei segnali
- Continuous: Blocchi per l'inserimento di funzioni di trasferimento

Simulink: Libreria Sources

Tra le varie Sources:

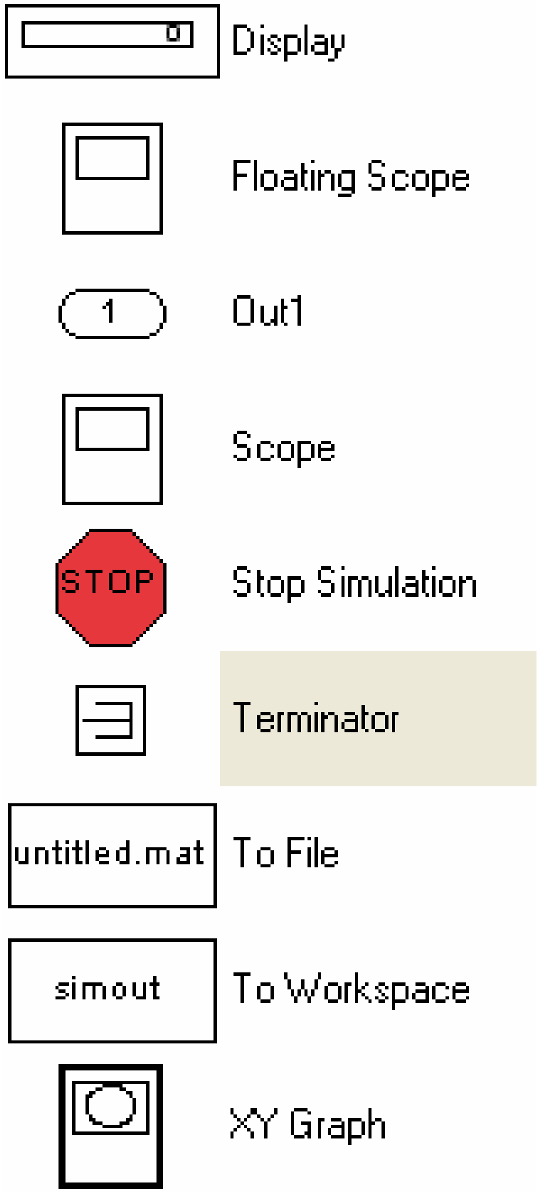
- **Constant:** genera un valore costante.
- **Step:** genera un gradino.
- **Ramp:** genera una rampa.
- **Sine wave:** genera una sinusoide.
- **From workspace:** il riferimento può essere generato in precedenza nel workspace e passato come `[tempo, valore]`, dove tempo e valore sono due vettori colonna di uguale lunghezza
- **Repeating sequence**
- **Clock:** Scandisce gli istanti di tempo della simulazione



Simulink: Libreria Sinks

Un insieme di strumenti che consente di visualizzare l'andamento di un segnale. I blocchi più importanti sono:

- **Scope:** Visualizza il segnale di ingresso in funzione del tempo.
- **XYGraph:** Genera un grafico del segnale connesso all'ingresso y (il secondo) in funzione di quello connesso all'ingresso x (il primo).
- **To Workspace:** Memorizza i valori del segnale connesso in una variabile MATLAB da usare poi per altri codici/funzioni.



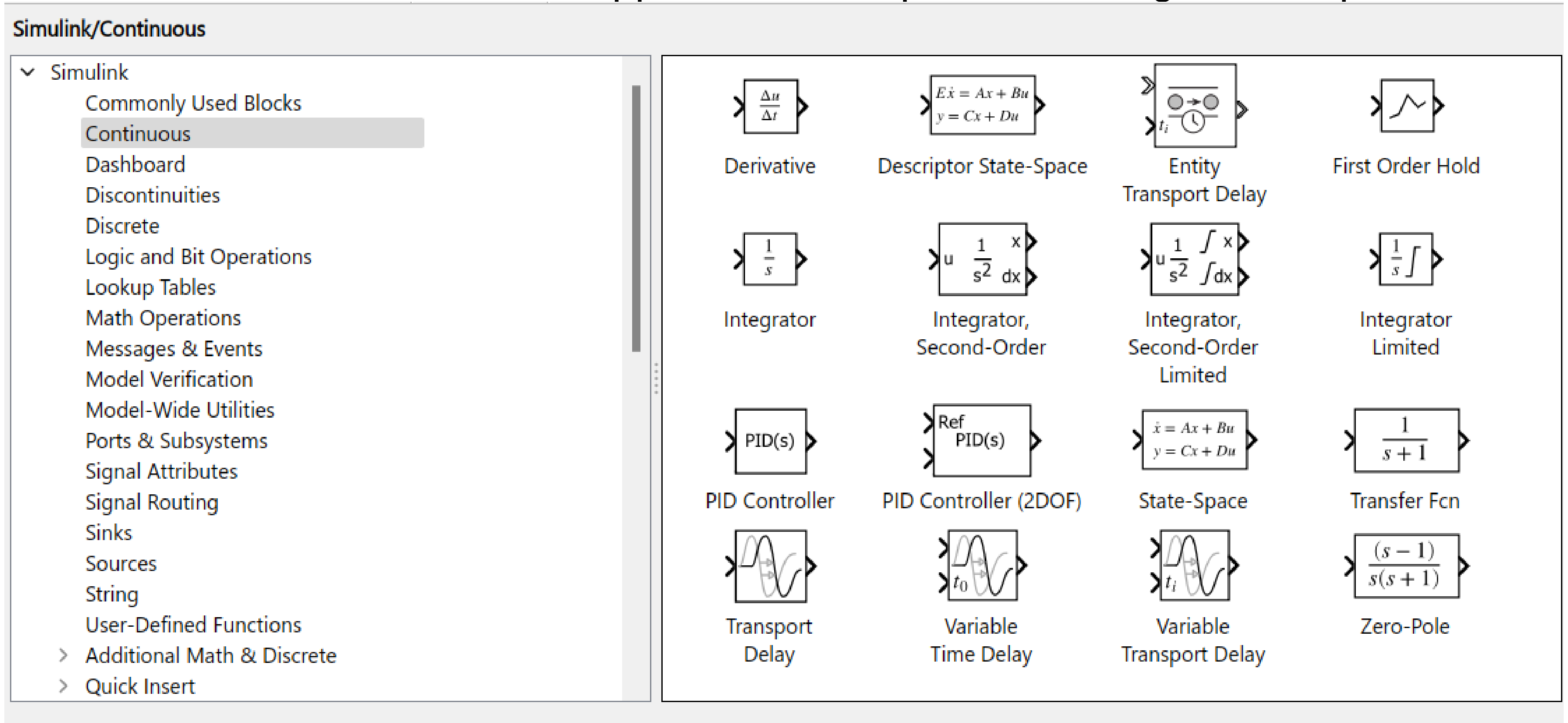
Per visualizzare l'andamento rispetto al tempo delle variabili, è necessario salvare in un'ulteriore variabile un vettore che scandisca gli istanti temporali della simulazione. Questo è possibile inserendo il blocco **clock** e collegandone l'uscita a un blocco **To Workspace** nello schema Simulink.

Simulink: Funzioni di trasferimento

Per inserire una funzione di trasferimento nello schema Simulink si utilizzano i blocchi presenti nella libreria **Continuous**:

• **Transfer Fcn:** consente di editare una funzione di trasferimento immettendo il numeratore e il denominatore. Numeratore e denominatore sono rappresentati da due vettori che esprimono i coefficienti, secondo potenze discendenti di *s*, del polinomio corrispondente.

• **Zero-Pole:** consente di editare una funzione di trasferimento specificando i suoi zeri e i suoi poli. Numeratore e denominatore sono rappresentati da due vettori i cui elementi rappresentano rispettivamente gli zeri e i poli della funzione di trasferimento.



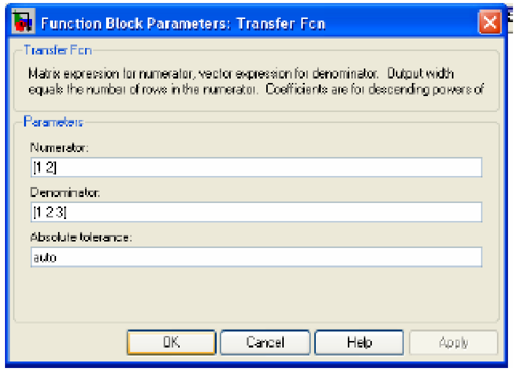
Se la funzione da inserire è un semplice integratore è già presente il blocco che lo implementa.

Simulink

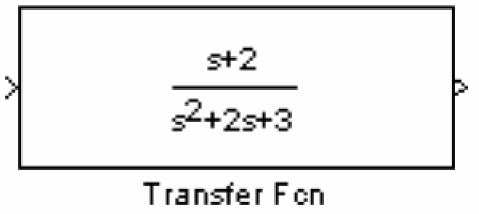
Se inseriamo nella maschera i vettori:

- *Numerator:* [1 2]
- *Denominator: [1 2 3]*

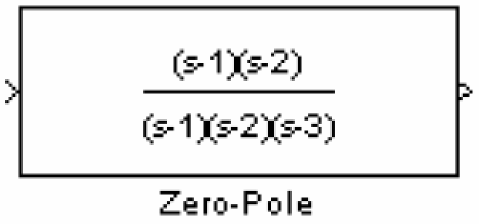
Otteniamo rispettivamente:



Transfer Fcn:



Zero-Pole:



Simulink

Ogni blocco prevede un set di uno o più parametri di configurazione (ad esempio, per il blocco «Gain» il valore del guadagno). I parametri di configurazione di un blocco sono settati all'interno di una finestra di configurazione alla quale si accede facendo doppio click sul blocco.

Realizzazione di un modello Simulink

- Importare nella pagina di lavoro i blocchi elementari Simulink necessari trascinandoli con il mouse dalla rispettiva libreria (drag and drop)
- Parametrizzare i blocchi Simulink nelle rispettive finestre di parametrizzazione, alle quali si accede facendo doppio click con il mouse sopra il blocco stesso.
- Collegare tra loro i blocchi Simulink tracciando le opportune linee di interconnessione in modo da realizzare le funzionalità desiderate

Simulink: Esempio

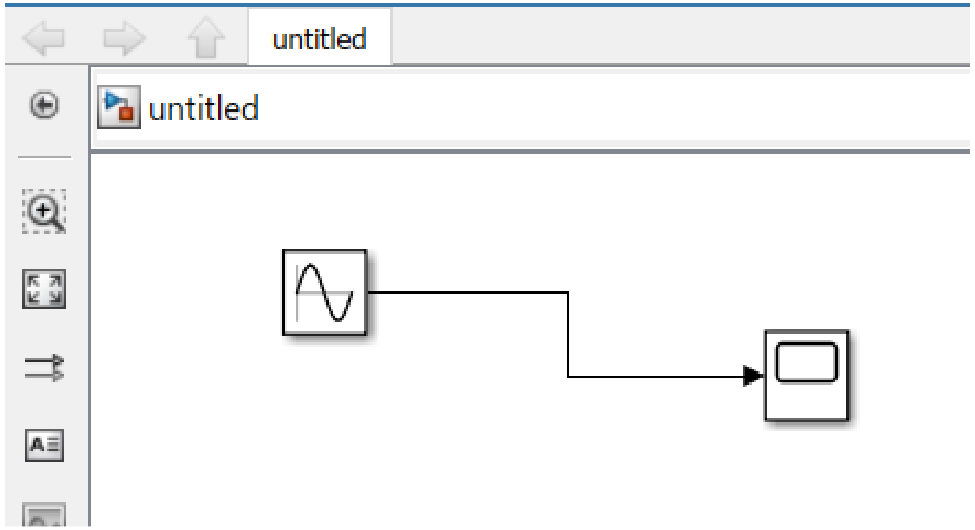
Costruzione e visualizzazione di un segnale sinusoidale
Sono sufficienti due blocchi elementari: un blocco che generi il segnale desiderato, ed un blocco che ne permetta la visualizzazione.
Il primo blocco lo troveremo anche nella libreria “Sources” (blocco Sine Wave).
Il secondo blocco (blocco Scope), si trova nella libreria dei Commonly Used Blocks ma anche nella libreria “Sinks”

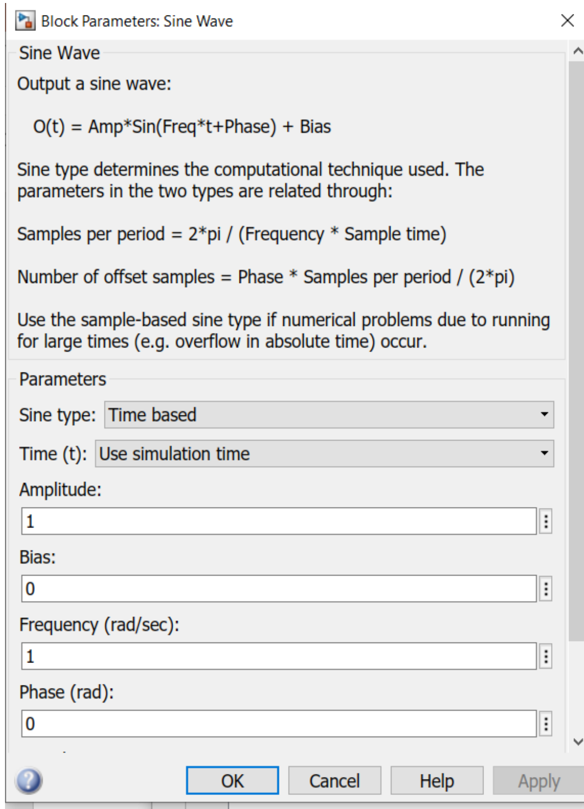
I blocchi necessari vanno importati nella pagina di lavoro Untitled con drag and drop dell’icona del blocco all’interno della pagina di lavoro Importiamo il blocco Sine Wave

Si vuole generare il segnale
s = $10+5\sin(t)$

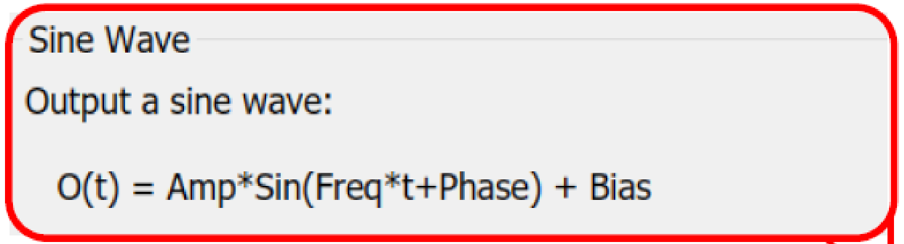
Per impostare i parametri della sinusoide fare doppio click sul blocco Si apre una finestra di dialogo all’interno della quale vanno impostati i suoi parametri di funzionamento

Impostare poi le connessioni





Simulink



Puo essere impostato anche un valore costante di **Bias** (v. formula in alto). Al termine della configurazione premere il tasto OK.

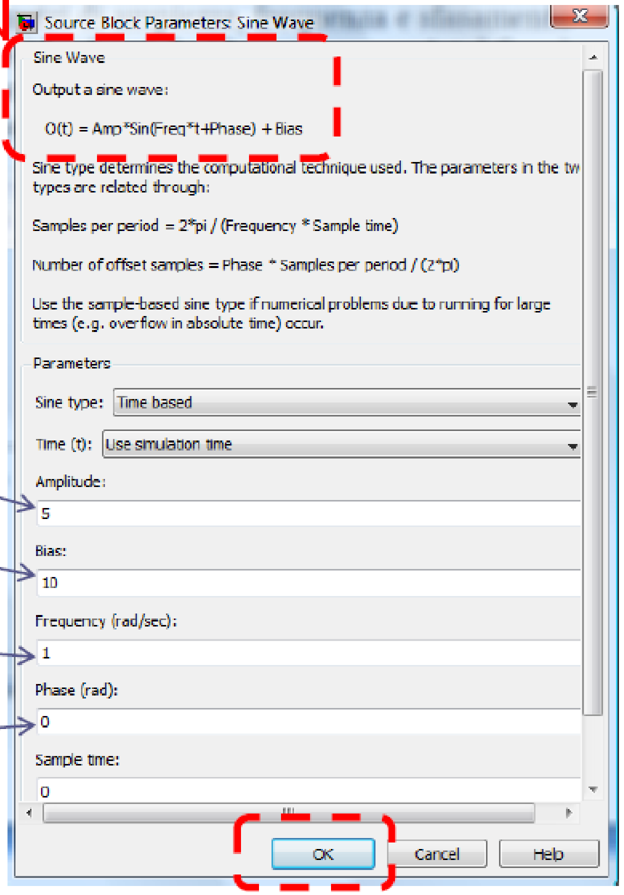
Ampiezza

Bias

Frequenza

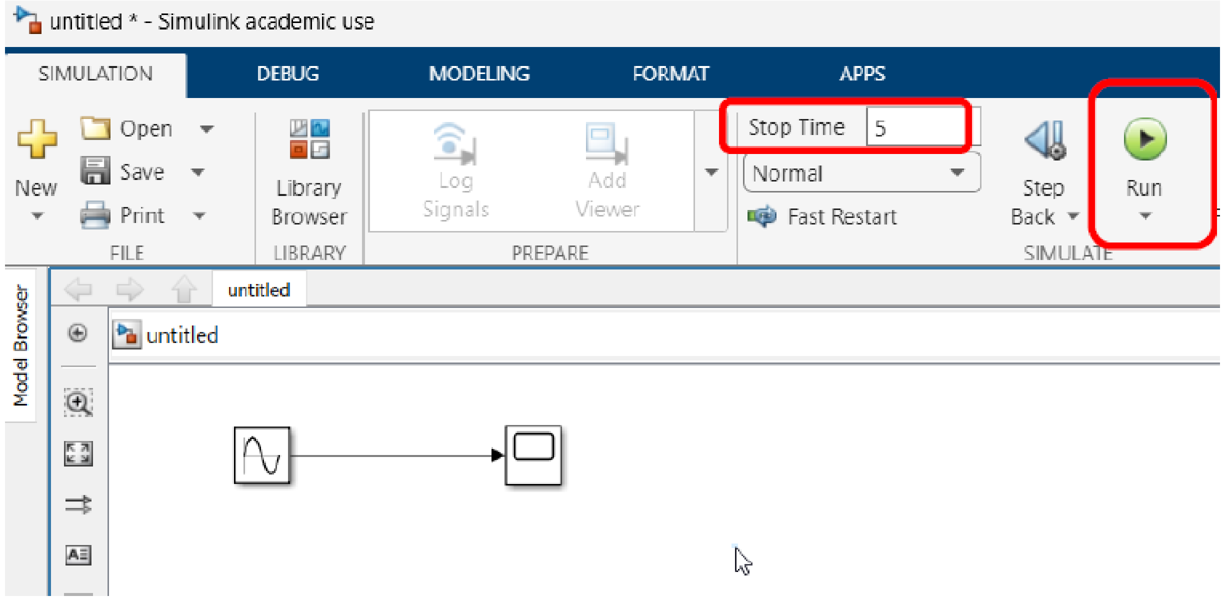
Sfasamento

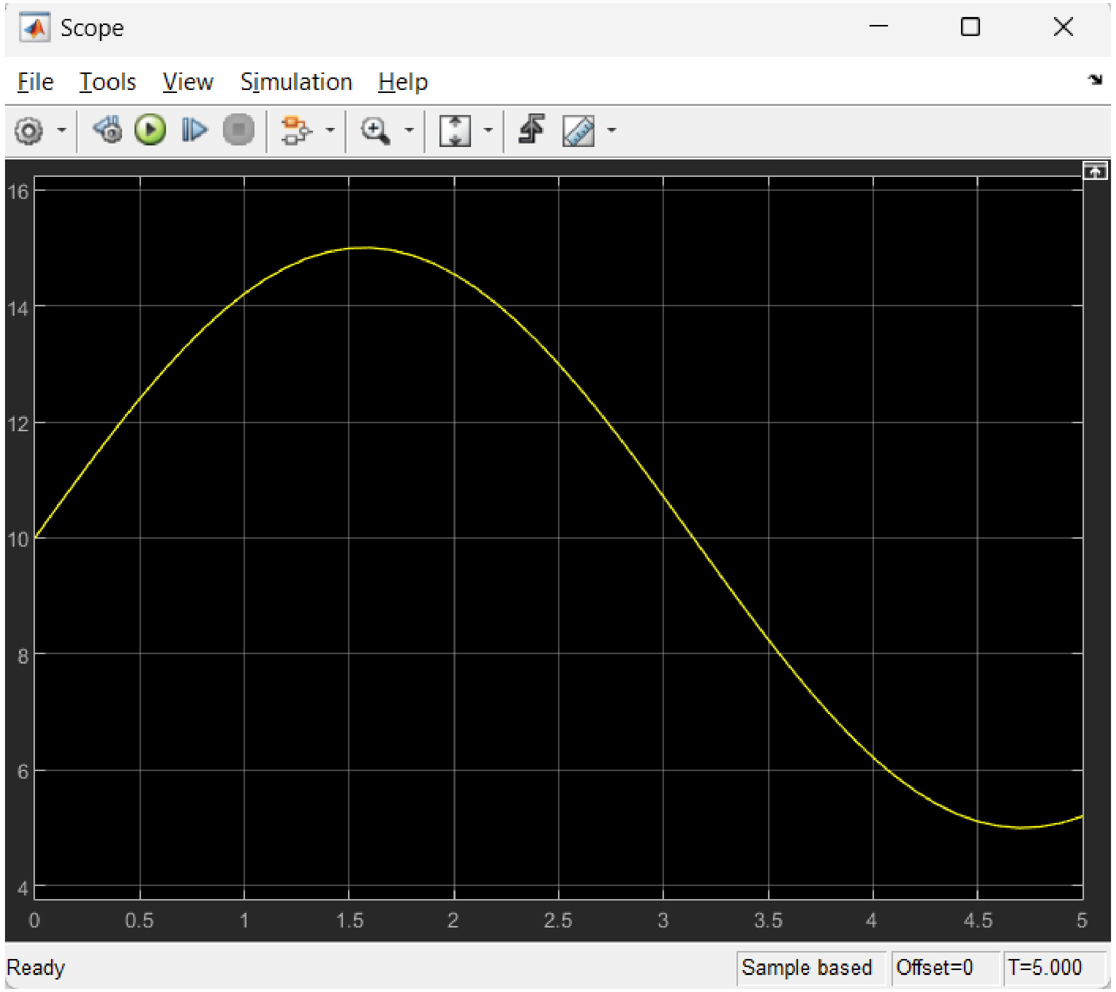
Tasto OK



Simulink

Impostare nella casella **Stop Time** la durata della simulazione (il valore di default è 10 secondi), e cliccare sul **pulsante Run** per avviare la simulazione.





Al termine della simulazione fare doppio click sul blocco Scope per visualizzare il segnale in una finestra grafica:

Simulink

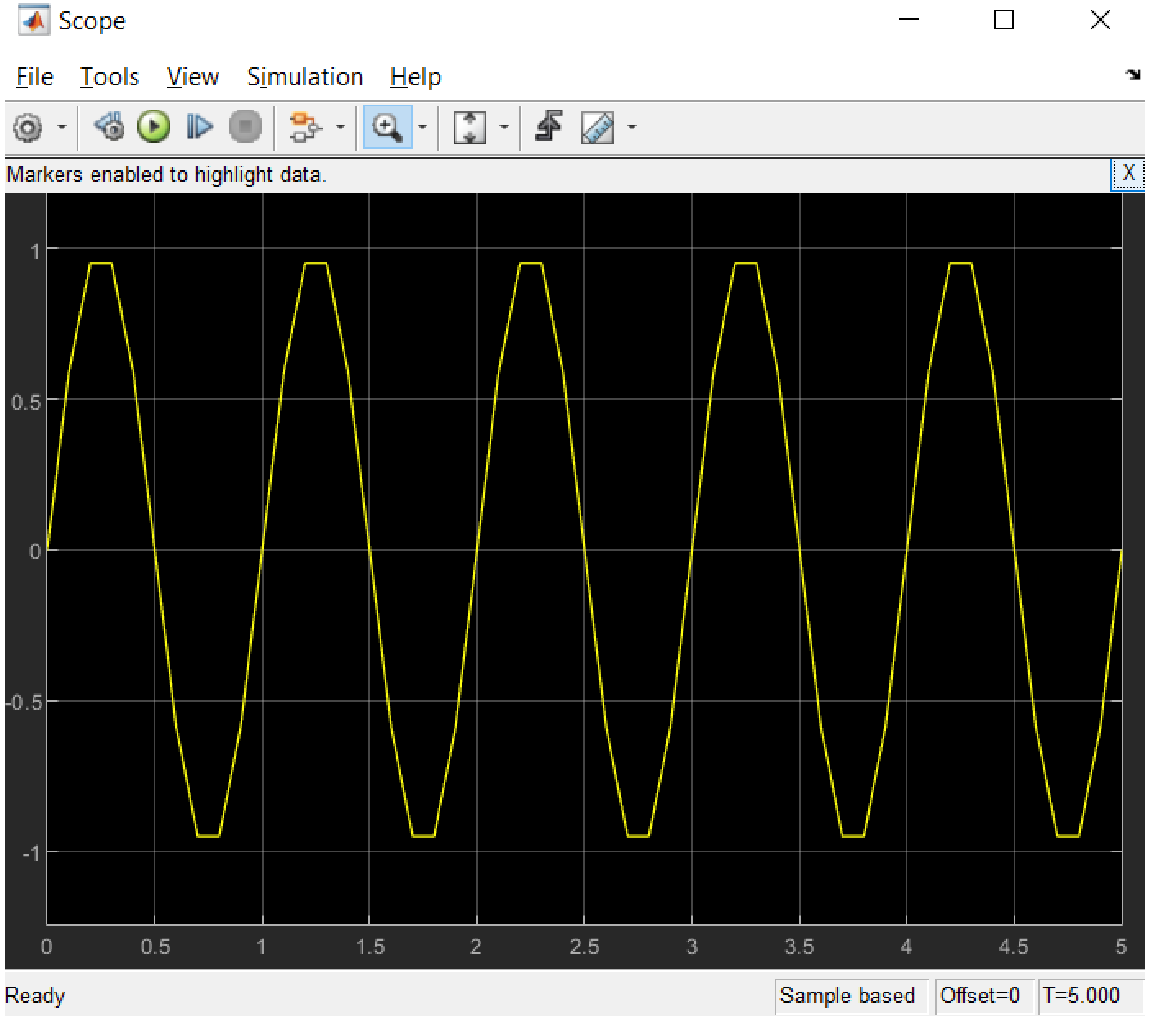


Grafico “spigoloso“
Il grafico è stato realizzato interpolando un numero di punti insufficiente

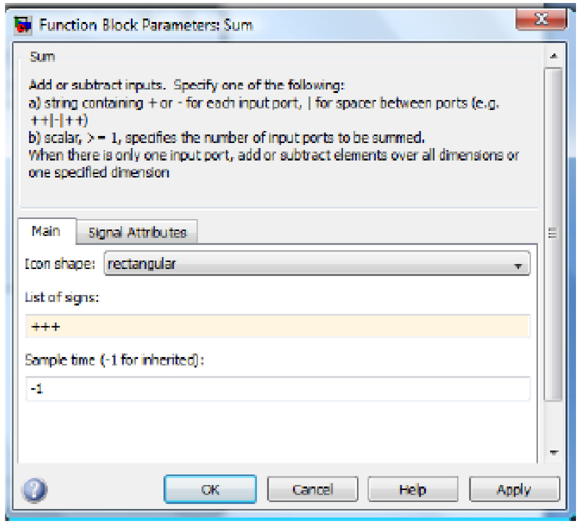
Si deve andare a modificare il “solver”, che definisce (fra le altre cose) il passo di discretizzazione temporale che viene impiegato nella esecuzione del modello

Fare click con il tasto destro in qualunque punto dello schema e scegliere dal menu «Model Configuration Parameters »

Simulink

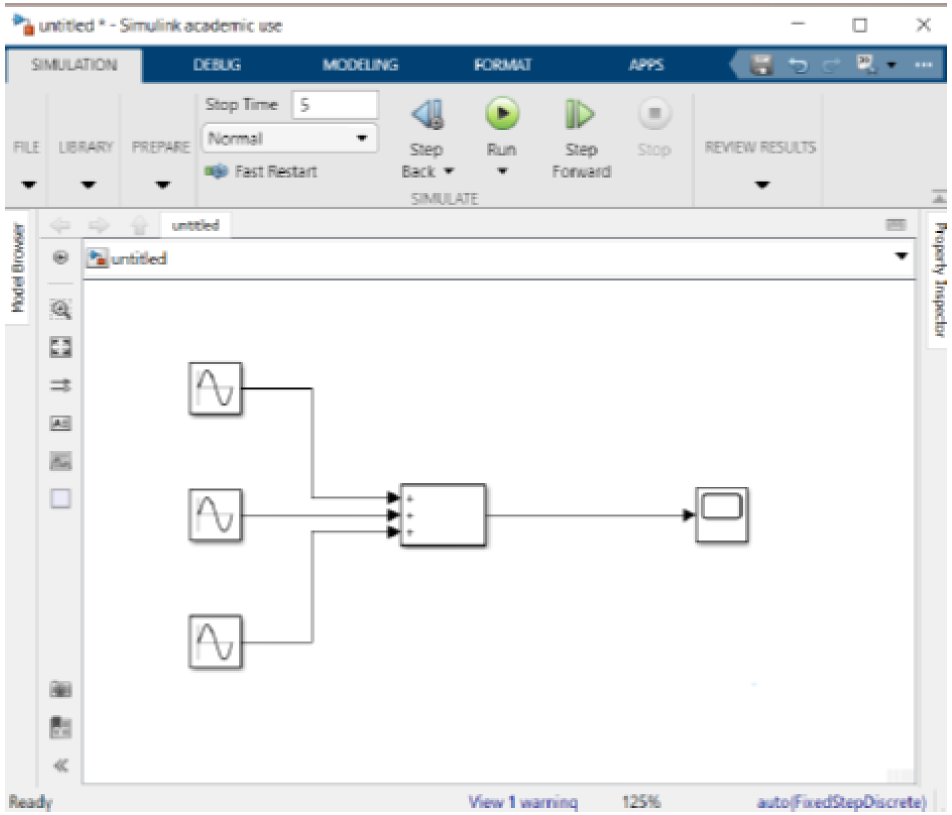
Per visualizzare un segnale costituito dalla **somma di tre sinusoidi** importiamo nella pagina di lavoro due nuove istanze del blocco elementare Sine Wave, ed importiamo anche un blocco che rappresenti un nodo sommatore (blocco Sum dalla libreria dei Commonly Used Blocks).

Il blocco Sum è parametrizzato per mezzo di una stringa (es. ++–+ –+) la cui lunghezza corrisponde al numero di segnali in ingresso al blocco mentre il segno + o - definisce se il corrispondente ingresso sia da sommare agli altri termini o da sottrarre.

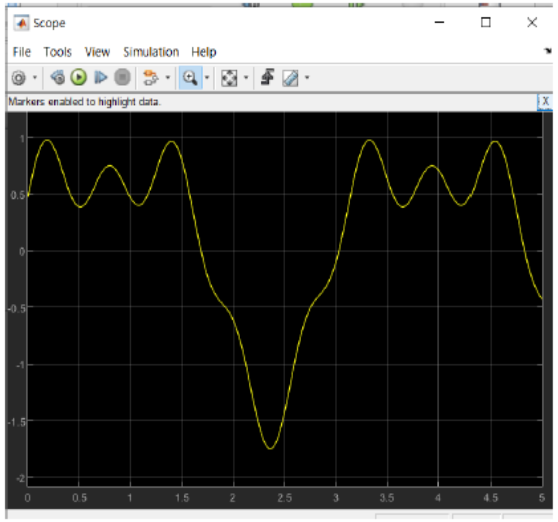


Scegliamo +++ e impostiamo la Icon shape in rectangular.
L’aspetto del blocco diventa

Simulink



	$sin(2t)$	$0.5\,cos(4t)$	$0.25\,sin(10t-0.1).$
Amplitude	1	0.5	0.25
Bias	0	0	0
Frequency	2	4	10
Phase	0	$pi/2$	-0.1



N.B. $cos(at) = sin(at + \pi/2)$

$$\sin(2t) + 0.5\cos(4t) + 0.25\sin(10t - 0.1).$$