

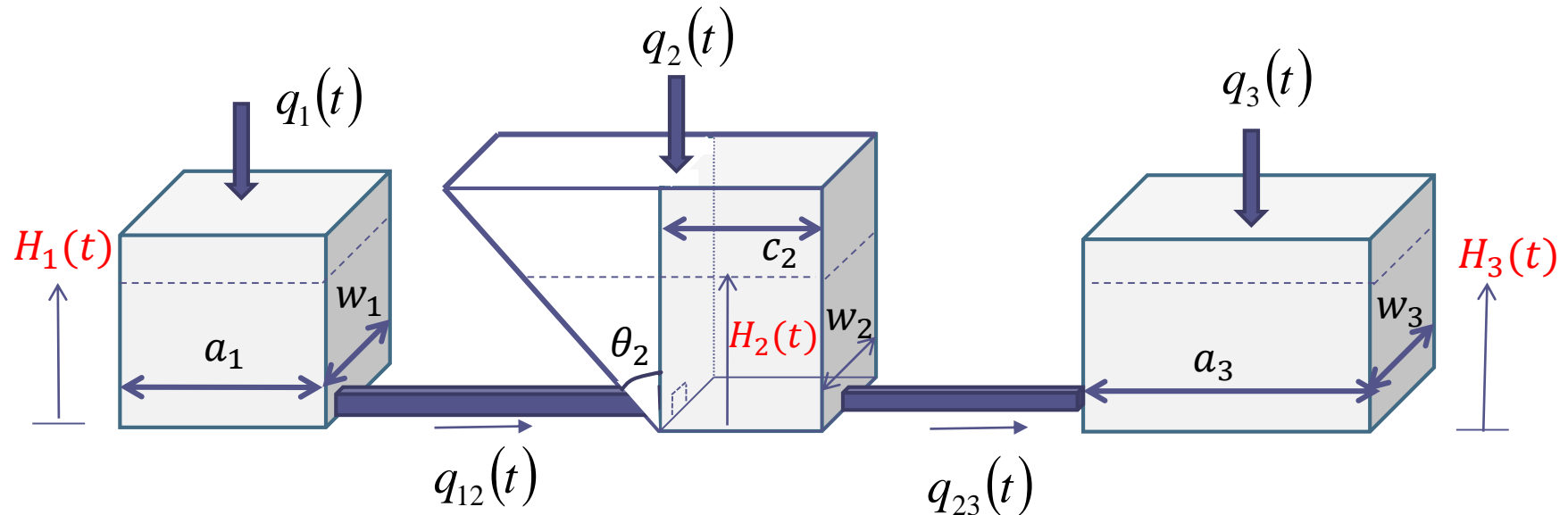
Simulazione dei Sistemi dinamici con Matlab-Simulink

Modellazione Simulink – parte seconda

Ing. Alessandro Pisano
apisano@unica.it

Serbatoi idraulici interconnessi

Modelliamo un sistema idraulico composto da tre serbatoi interconnessi da due pipeline



$H_1(t), H_2(t), H_3(t)$

Livelli del liquido nei 3 serbatoi

$q_1(t), q_2(t), q_3(t)$

Portate volumetriche esterne in ingresso nei 3 serbatoi

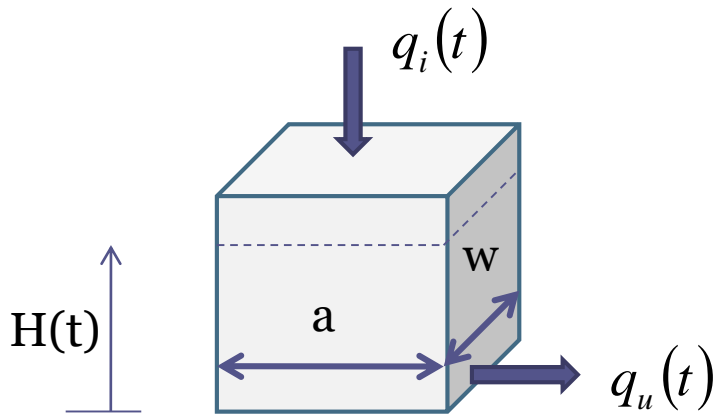
$q_{12}(t)$ ($q_{23}(t)$)

Portata volumetrica che fluisce fra i serbatoi 1 e 2 (**2 e 3**)

$a_1, w_1, c_2, w_2, \theta_2, a_3, w_3$

Dimensioni geometriche dei serbatoi

Serbatoio a sezione costante rettangolare



$V(t)$ = volume di liquido contenuto nel serbatoio (in m^3)

$$V(t) = awH(t)$$

$H(t)$ = livello del liquido nel serbatoio (in m)

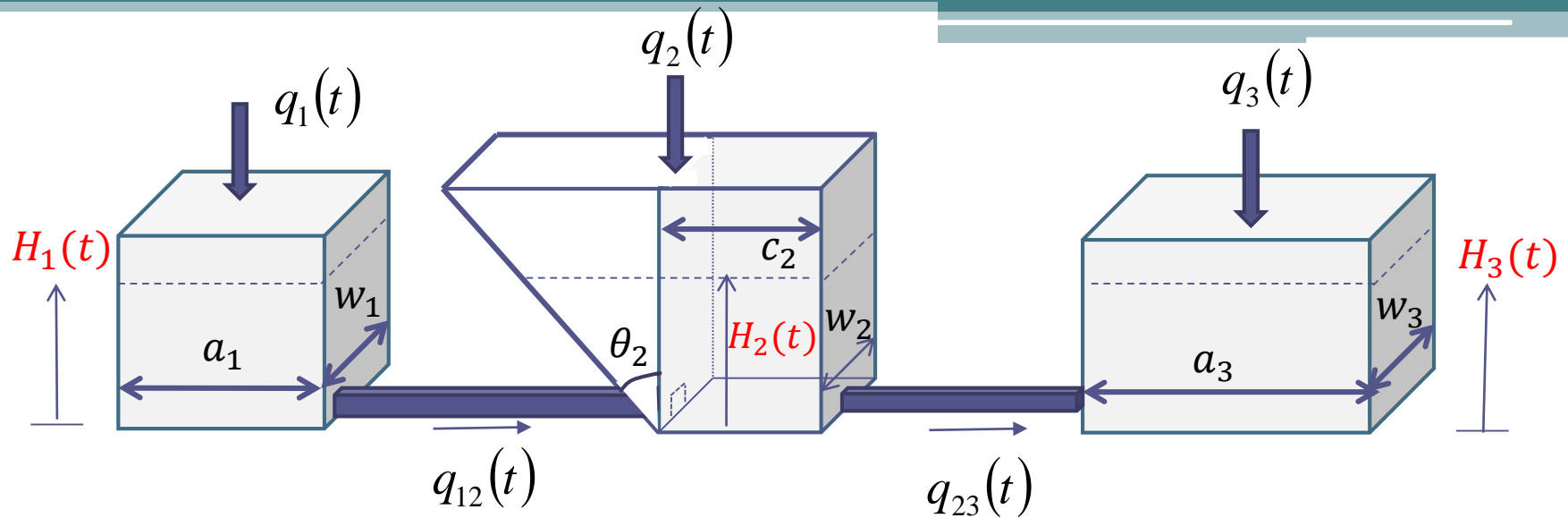
$q_i(t)$ $q_u(t)$ portate volumetriche in ingresso ed in uscita dal serbatoio (in m^3/sec)

Conservazione
della massa

$$\dot{V}(t) = q_i(t) - q_u(t)$$

$$\dot{V}(t) = aw\dot{H}(t) = q_i(t) - q_u(t)$$

$$\dot{H}(t) = \frac{1}{aw} q_i(t) - \frac{1}{aw} q_u(t)$$



$$\dot{H}_1(t) = \frac{1}{a_1 w_1} [q_1(t) - q_{12}(t)]$$

$$\dot{H}_2(t) = \frac{1}{\tan(\theta_2) w_2 H_2(t) + c_2 w_2} [q_2(t) + q_{12}(t) - q_{23}(t)]$$

**Modello in
forma esplicita**

$$\dot{H}_3(t) = \frac{1}{a_3 w_3} [q_3(t) + q_{23}(t)]$$

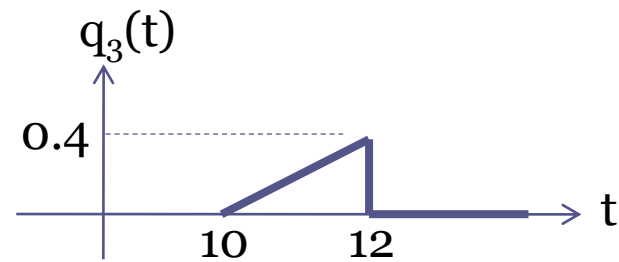
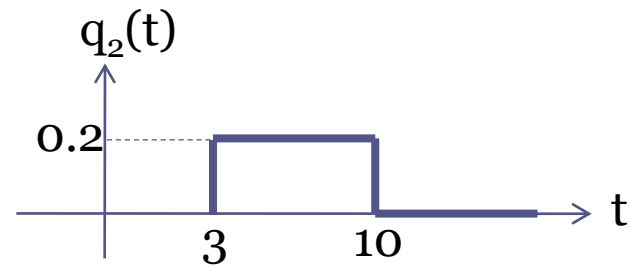
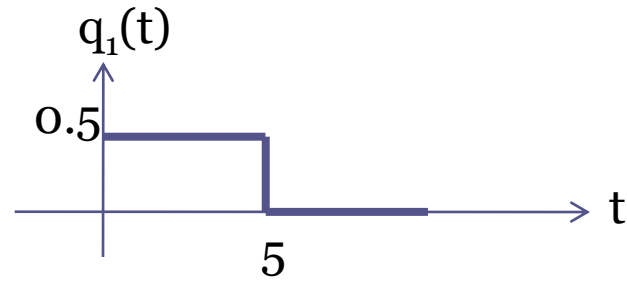
$$q_{12}(t) = C_{12} \sqrt{|H_1(t) - H_2(t)|} \cdot \text{sign}[H_1(t) - H_2(t)]$$

$$q_{23}(t) = C_{23} \sqrt{|H_2(t) - H_3(t)|} \cdot \text{sign}[H_2(t) - H_3(t)]$$

C_{12}, C_{23}

Coefficienti di efflusso

Portate volumetriche in ingresso



File: Serbatoi_dati.m

```
%% Parametrizzazione del modello
% Parametri serbatoio #1
a1=5;
w1=1;

% Parametri serbatoio #2
c2=3;
w2=1;
teta2=pi/4;

% Parametri serbatoio #2
a3=4;
w3=2;

% Coefficienti di efflusso
C12=0.5;
C23=0.6;

% Condizioni iniziali
H10=1;
H20=2;
H30=3;
```

**Eseguiamo lo script in
modo da creare nel
workspace le
corrispondenti variabili**

Apriamo un modello Simulink in bianco, e configuriamo il solutore come segue (oppure si apra il template creato in precedenza che già possiede tali impostazioni)

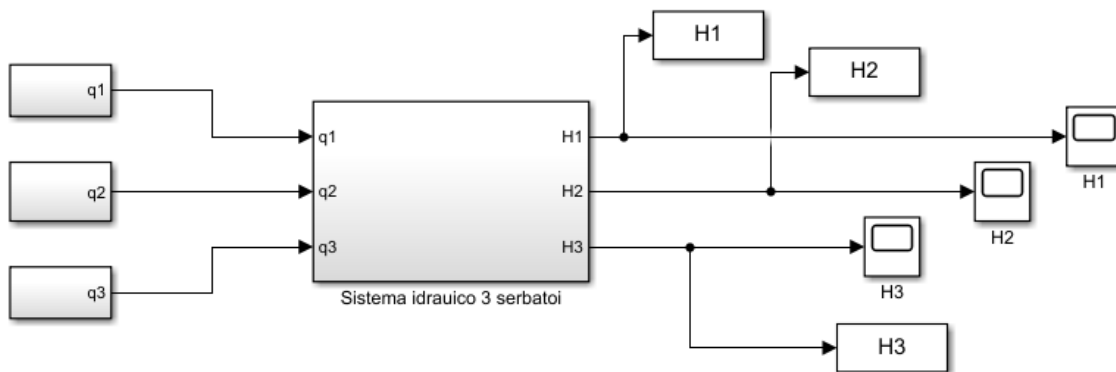
Type: fixed step;

Solver: auto (Automatic Solver selection);

Fixed step size: 0.01 s)

Desideriamo strutturare il modello mediante un Subsystem che riceva in ingresso le portate $q_1(t)$, $q_2(t)$, $q_3(t)$ e fornisca in uscita i 3 livelli $H_1(t)$, $H_2(t)$, $H_3(t)$

Obiettivo:



Anche le 3 portate $q_1(t)$, $q_2(t)$, $q_3(t)$ desideriamo che vengano generate all'interno di appositi sottosistemi.

Descriviamo una procedura differente per realizzare un modello contenente dei Subsystems. Importiamo nella pagina di lavoro un blocco «Subsystem» dalla Libreria dei «Commonly Used Blocks»

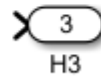
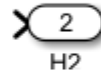
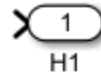
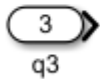
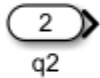


(cliccare il tasto destro del mouse sul Subsystem e disselezionare la voce «Content preview» dal menu format)

Si tratta di un sottosistema già pronto, che ha di default un ingresso ed un'uscita.

Trasformiamolo in un subsystem con tre ingressi e tre uscite **incrementando il numero di blocchi «In» e «Out» presenti all'interno del Subsystem**. Inseriamo anche opportune etichette, come già fatto in precedenza.

Contenuto del Subsystem



Aspetto del Subsystem



Abbiamo creato lo «scheletro» del Subsystem. Ora bisogna completarne il contenuto in modo che questo implementi il sistema di equazioni che definiscono il modello del sistema idraulico a 3 serbatoi.

Importiamo all'interno del Subsystem **tre blocchi integratori**, e disponiamoli come in figura (si impostino anche le relative condizioni iniziali mediante le variabili H10, H20 ed H30). Faremo in modo che alle porte di uscita dei tre integratori vengano generati i tre livelli $H_1(t)$, $H_2(t)$, $H_3(t)$



Iniziamo con il costruire i segnali $q_{12}(t)$ e $q_{23}(t)$

$$q_{12}(t) = C_{12} \sqrt{|H_1(t) - H_2(t)|} \cdot \text{sign}[H_1(t) - H_2(t)]$$

$$q_{23}(t) = C_{23} \sqrt{|H_2(t) - H_3(t)|} \cdot \text{sign}[H_2(t) - H_3(t)]$$

Utilizziamo un blocco chiamato «**Fcn**». Tale blocco è stato dismesso a partire dalla versione R2020b ma risulta estremamente utile per semplificare la struttura dei modelli Simulink. Il blocco è disponibile all'interno del file `Blocco_Fcn.slx`

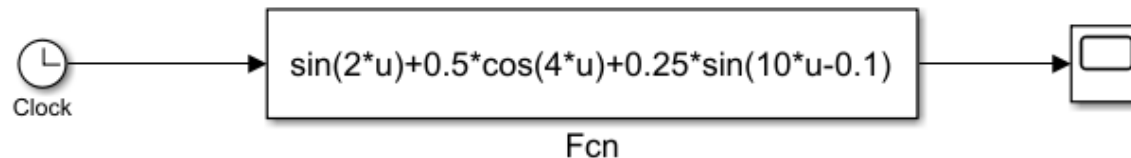
Tale blocco riceve in ingresso un segnale (scalare o vettoriale) e consente di scrivere al suo interno una **espressione** contenente funzioni Matlab che viene applicata al segnale di ingresso ed il cui risultato viene restituito sul terminale di uscita sotto forma di una quantità **scalare**.

L'espressione può includere uno o più dei seguenti elementi

- **u** — The input to the block. **If u is a vector, u(i) represents the ith element of the vector**; u(1) or u alone represents the first element.
- **Numeric constants** . (pi, eps,).
- **Arithmetic operators** (+ - * / ^).
- **Relational operators** (== != > < >= <=) — The expression returns 1 if the relation is true; otherwise, it returns 0.
- **Logical operators** (& & || !) — The expression returns 1 if the relation is true; otherwise, it returns 0.
- **Parentheses**.
- **Mathematical functions** —
abs, acos, asin, atan, atan2, ceil, cos, cosh, exp, floor, hypot, log, log10, power, rem, sgn (equivalent to [sign](#) in MATLAB), sin, sinh, sqrt, tan, and tanh.
- **Workspace variables** — Matrix or vector elements must be specifically referenced (e.g., A(1,1) instead of A for the first element in the matrix).

Nella sua modalità di impiego «standard» il blocco riceve in ingresso un segnale scalare che all'interno dell'espressione viene denotato con «u»

Per costruire la forma d'onda $\sin(2t) + 0.5 \cos(4t) + 0.25 \sin(10t - 0.1)$ (esempio trattato in precedenza) si può utilizzare la seguente configurazione

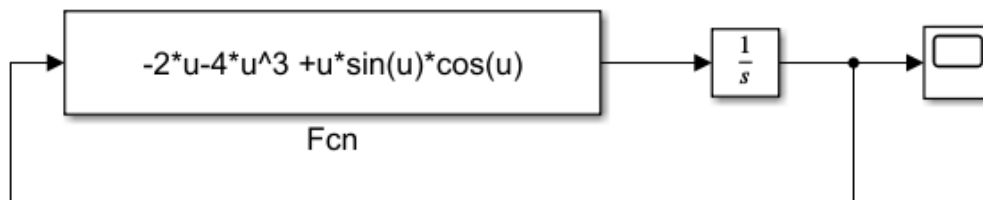


Per risolvere il problema di Cauchy

$$\dot{x}(t) = -2x(t) - 4x^3(t) + x(t) \sin(x(t)) \cos(x(t))$$

$$x(0) = 1$$

si può utilizzare la seguente configurazione



File: Cauchy.slx

E' anche possibile passare in ingresso al blocco Fcn un **segnale «vettoriale»** cioè un array di segnali $s_1(t), s_2(t), \dots, s_n(t)$

$$\begin{bmatrix} s_1(t) \\ s_2(t) \\ \vdots \\ s_n(t) \end{bmatrix}$$

Questa è la modalità di impiego che ci serve nel presente esempio per generare mediante il blocco Fcn i segnali

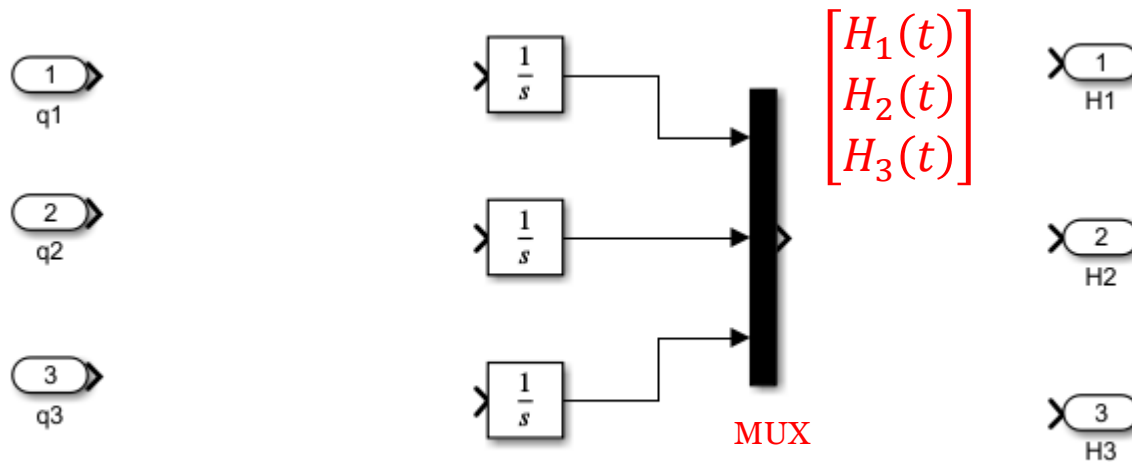
$$q_{12}(t) = C_{12} \sqrt{|H_1(t) - H_2(t)|} \cdot \text{sign}[H_1(t) - H_2(t)]$$

$$q_{23}(t) = C_{23} \sqrt{|H_2(t) - H_3(t)|} \cdot \text{sign}[H_2(t) - H_3(t)]$$

Creiamo all'interno del Subsystem
un array di segnali che contenga i tre
livelli dei serbatoi

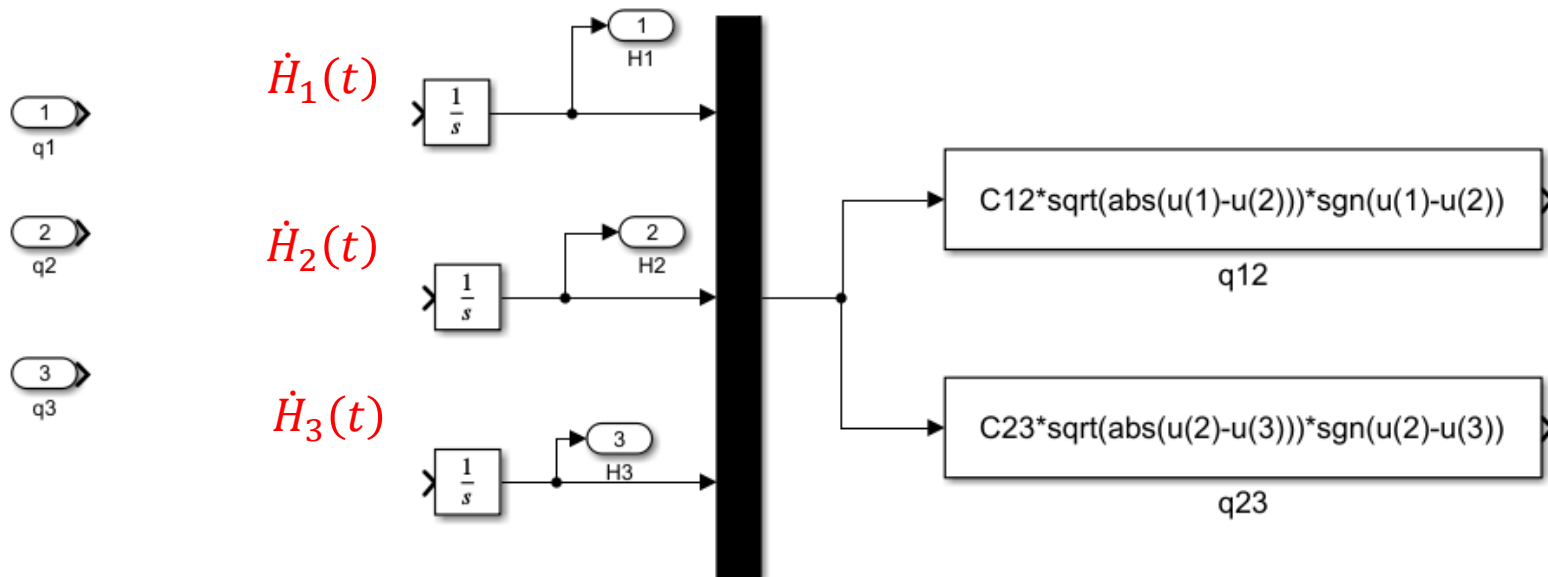
$$\begin{bmatrix} H_1(t) \\ H_2(t) \\ H_3(t) \end{bmatrix}$$

La creazione di un array di segnali si fa mediante il blocco Simulink «Mux» (libreria «Commonly Used Blocks»)



Il blocco «Mux» ha di default due terminali di ingresso. Lo si configuri in modo da poter accorpare fra loro tre segnali (Number of inputs: 3).

Ora possiamo applicare il segnale vettoriale in ingresso a due istanze del blocco Fcn, e scrivere le relative espressioni. **All'interno delle espressioni, le tre componenti del vettore in ingresso sono indicate come $u(1)$, $u(2)$, $u(3)$**



$$q_{12}(t) = C_{12} \sqrt{|H_1(t) - H_2(t)|} \cdot \text{sign}[H_1(t) - H_2(t)]$$

$$q_{23}(t) = C_{23} \sqrt{|H_2(t) - H_3(t)|} \cdot \text{sign}[H_2(t) - H_3(t)]$$

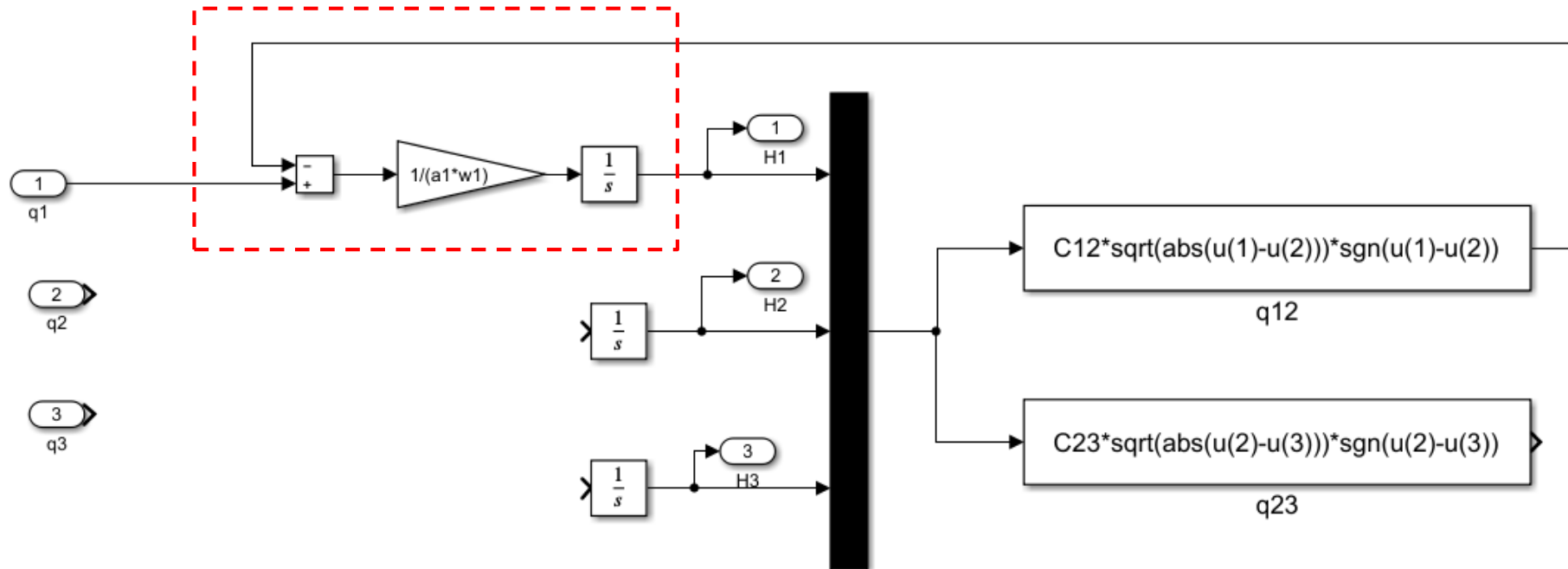
A questo punto possiamo costruire i segnali di ingresso ai vari integratori secondo le equazioni del modello in forma esplicita

$$\ddot{H}_1(t) = \frac{1}{a_1 w_1} [q_1(t) - q_{12}(t)]$$

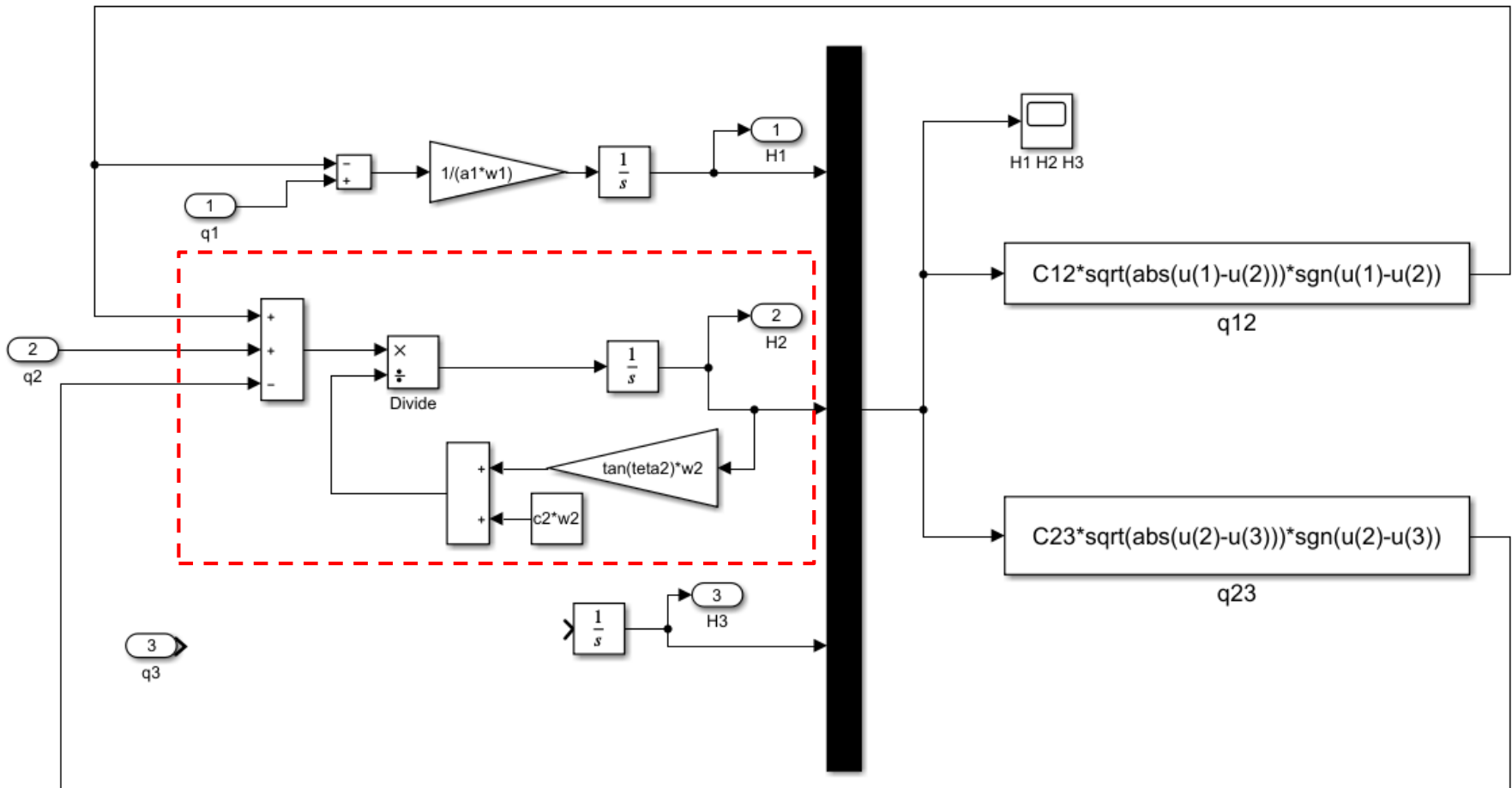
$$\ddot{H}_2(t) = \frac{1}{\tan(\theta_2) w_2 H_2(t) + c_2 w_2} [q_2(t) + q_{12}(t) - q_{23}(t)]$$

$$\ddot{H}_3(t) = \frac{1}{a_3 w_3} [q_3(t) + q_{23}(t)]$$

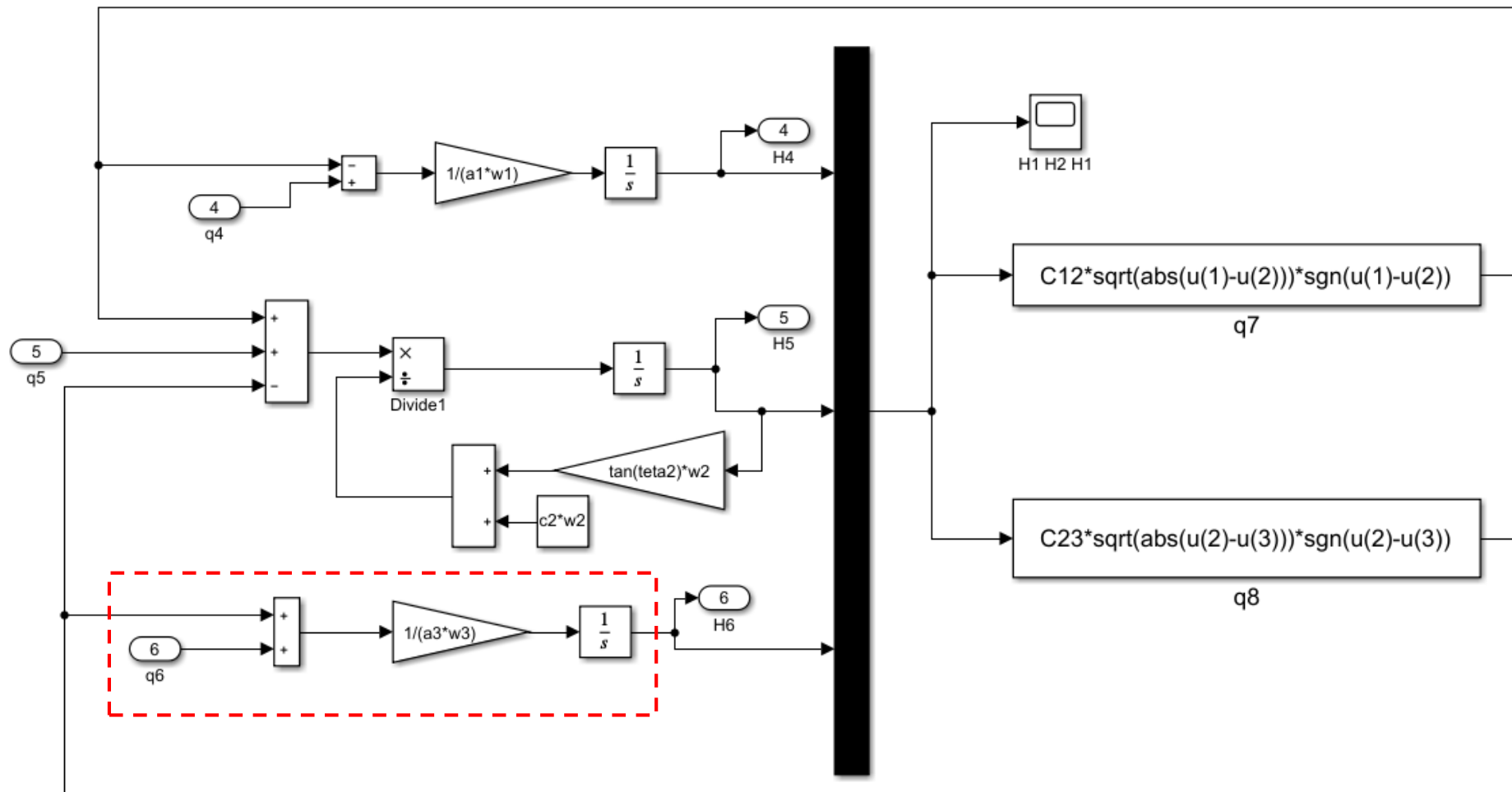
$$\dot{H}_1(t) = \frac{1}{a_1 w_1} [q_1(t) - q_{12}(t)]$$



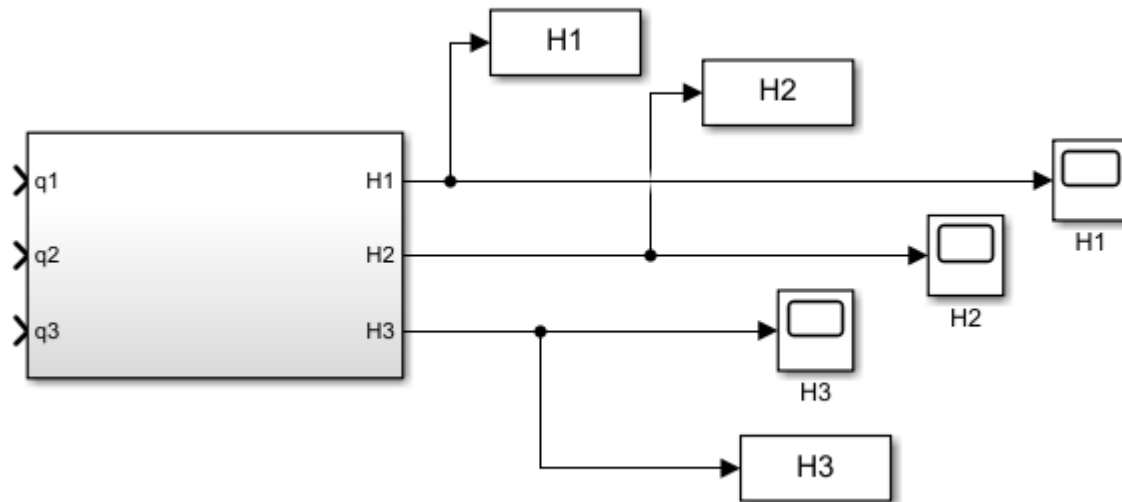
$$\dot{H}_2(t) = \frac{1}{\tan(\theta_2) w_2 H_2(t) + c_2 w_2} [q_2(t) + q_{12}(t) - q_{23}(t)]$$



$$\dot{H}_3(t) = \frac{1}{a_3 w_3} [q_3(t) + q_{23}(t)]$$



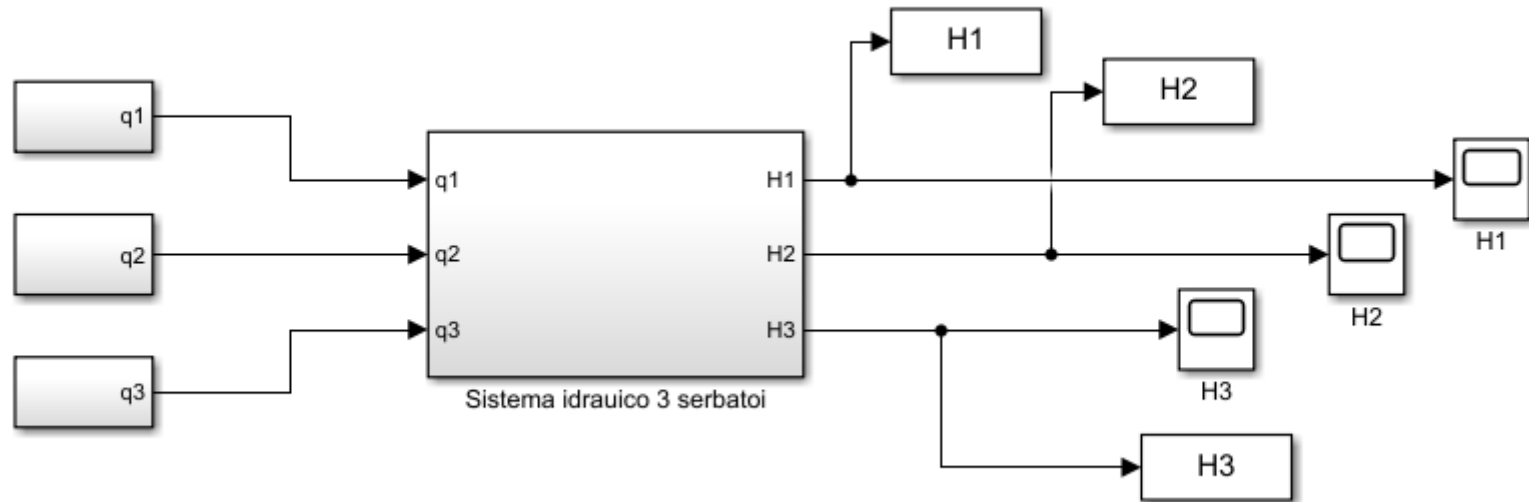
Modello completo all'interno della pagina di lavoro. Sono stati collegati ai tre terminali di uscita del Subsystem i relativi blocchi **Scope** e tre blocchi «**To workspace**» per salvare nel workspace i tre livelli dei serbatoi



Per poter avviare la simulazione devono essere generati i tre segnali $q_1(t)$, $q_2(t)$, $q_3(t)$ che rappresentano le portate esterne in ingresso ai serbatoi.

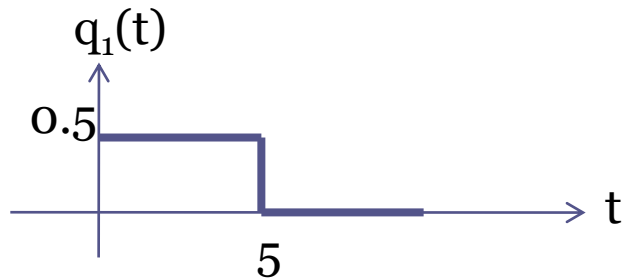
Creiamo tre Subsystem preposti alla generazione di tali segnali.

Come già fatto in precedenza, importiamo tre blocchi «Subsystem» già pronti dalla libreria dei «Commonly used blocks» e modifichiamone il contenuto. In particolare **rimuoviamo i blocchi «In» presenti al loro interno** in modo che i tre Subsystem abbiano unicamente un terminale di uscita. Diamo ai terminali di uscita dei tre Subsystem i nomi q1, q2, q3.

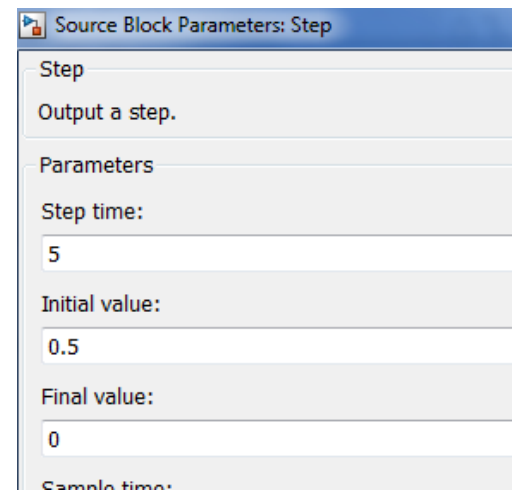
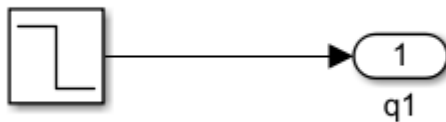


Ora inseriamo all'interno dei Subsystem i blocchi necessari per generare le tre forme d'onda desiderate per i segnali $q_1(t)$, $q_2(t)$, $q_3(t)$

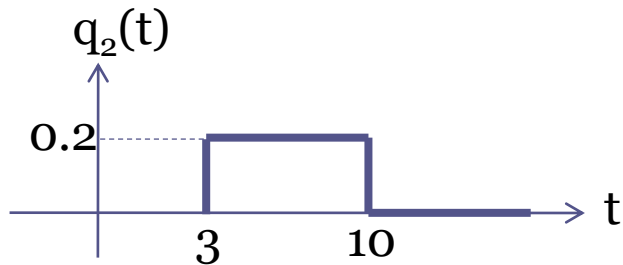
Segnale $q_1(t)$



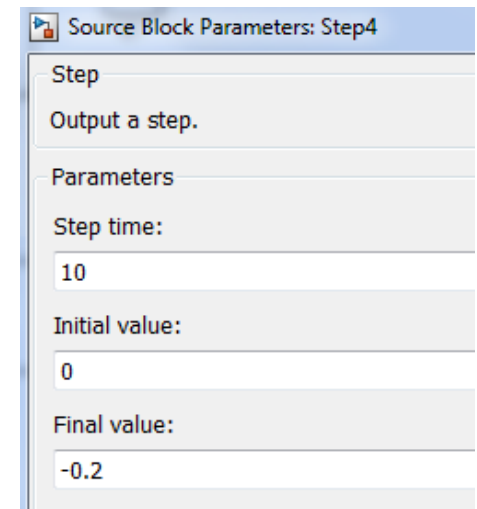
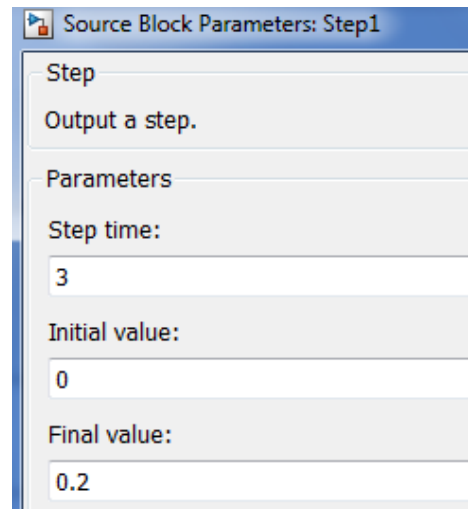
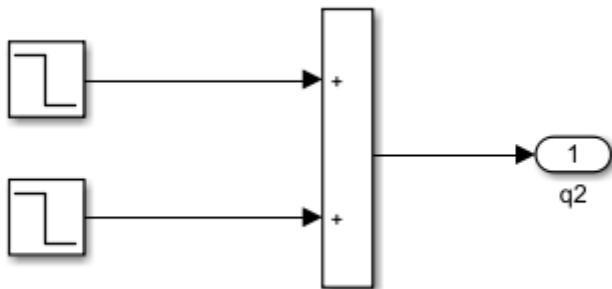
E' sufficiente un singolo blocco «Step» (libreria: Sources, blocco già spiegato in precedenza).



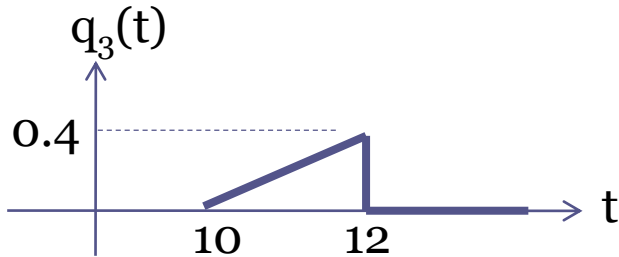
Segnale $q_2(t)$



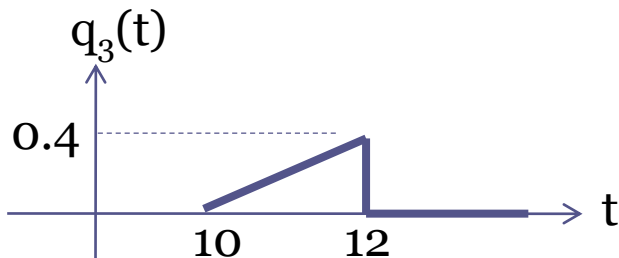
Possiamo generare questo segnale sommando fra loro le uscite di due blocchi Step parametrizzati opportunamente



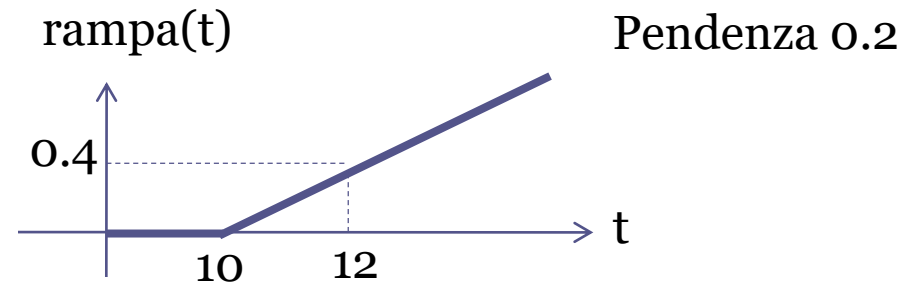
Segnale $q_3(t)$



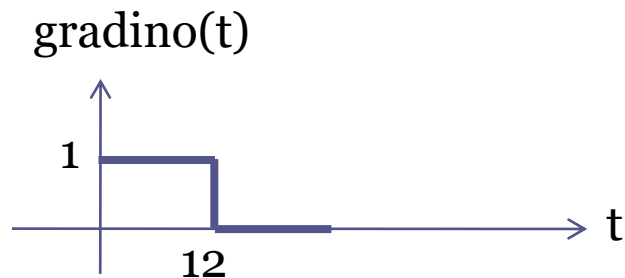
Possiamo generare questo segnale **moltiplicando fra loro un segnale a rampa** (blocco «Ramp», libreria Sources) **ed un segnale a gradino** (blocco Step)



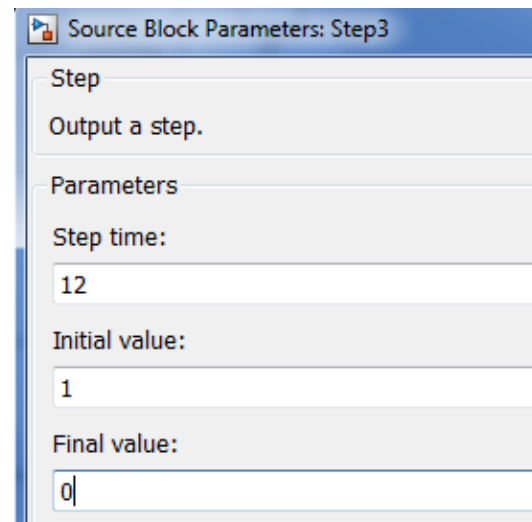
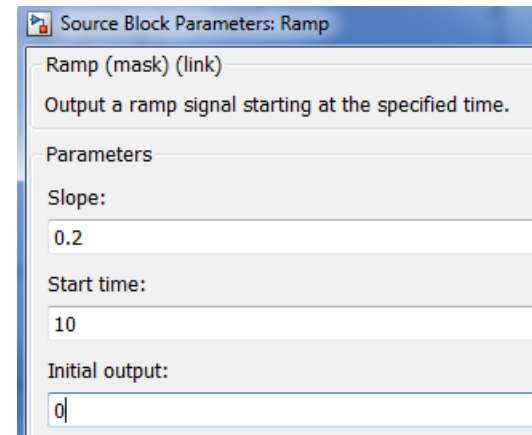
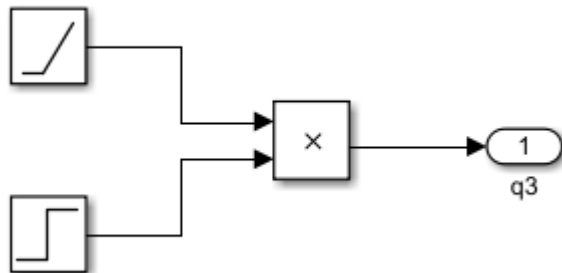
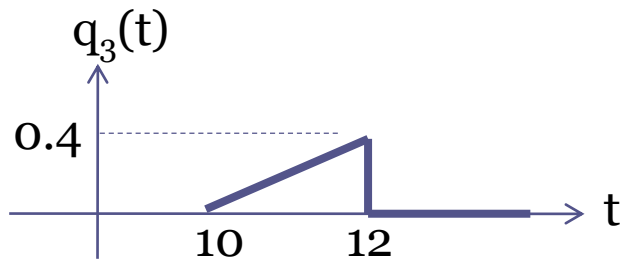
=



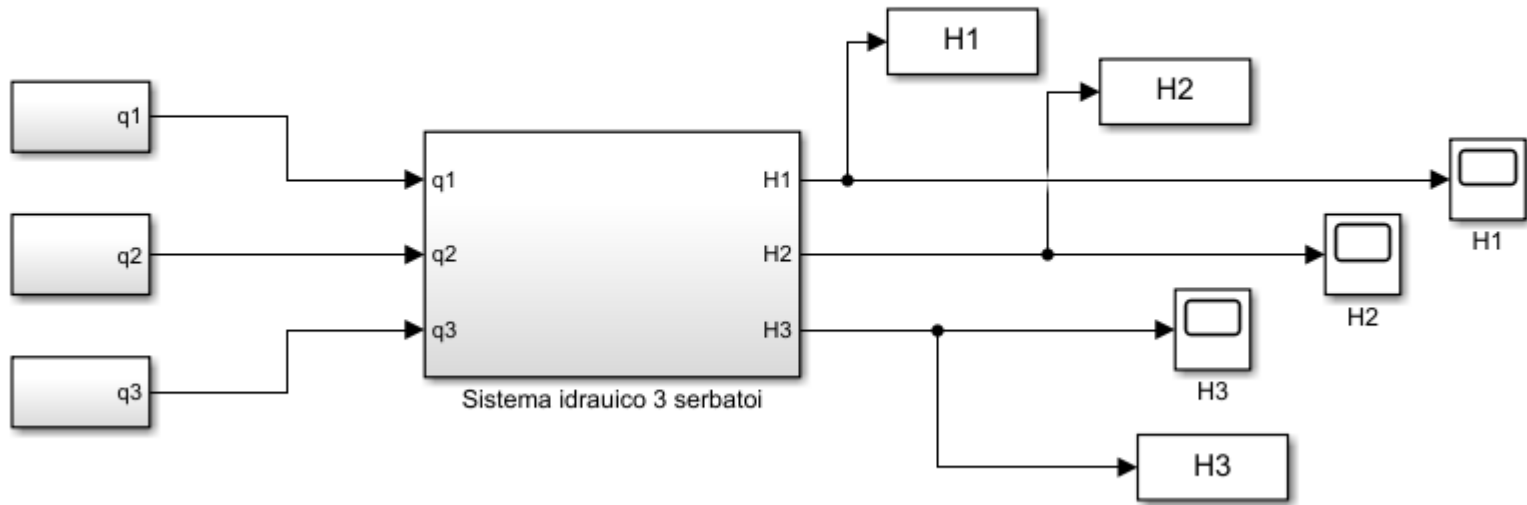
×



Segnale $q_3(t)$



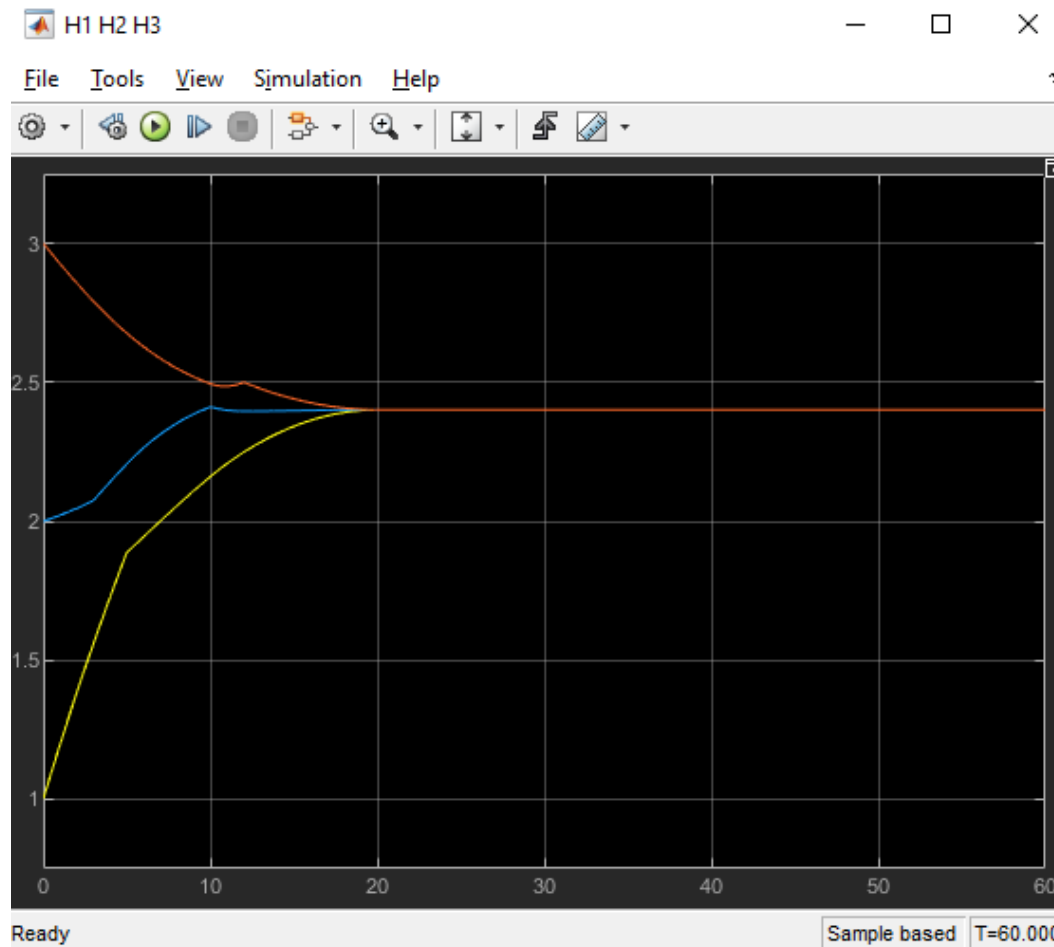
Modello completo



File: Serbatoi_v01.slx

Ora si può fare il run del modello (impostare come durata della simulazione 60 secondi) e visualizzare i segnali $H_1(t)$, $H_2(t)$, $H_3(t)$.

Se si applica in ingresso ad un blocco Scope un segnale vettoriale (si colleghi l'uscita del blocco Mux all'ingresso di un blocco Scope) vengono visualizzati sovrapposti tutti i segnali



Nella Section finale dello script `Serbatoi_dati.m` sono inserite le istruzioni che attivano il run del modello (comando `sim`) e creano vari grafici con i risultati

```
%% Run del modello e creazione grafici
```

```
sim('Serbatoi_v01')
```

```
figure(1)
```

```
plot(H1, 'LineWidth', 2)
```

```
grid,
```

```
xlabel('Tempo [s]')
```

```
title('Livello H1')
```

```
figure(2)
```

```
plot(H2, 'LineWidth', 2)
```

```
grid,
```

```
xlabel('Tempo [s]')
```

```
title('Livello H2')
```

```
figure(3)
```

```
plot(H3, 'LineWidth', 2)
```

```
grid,
```

```
xlabel('Tempo [s]')
```

```
title('Livello H3')
```

```
figure(4)
```

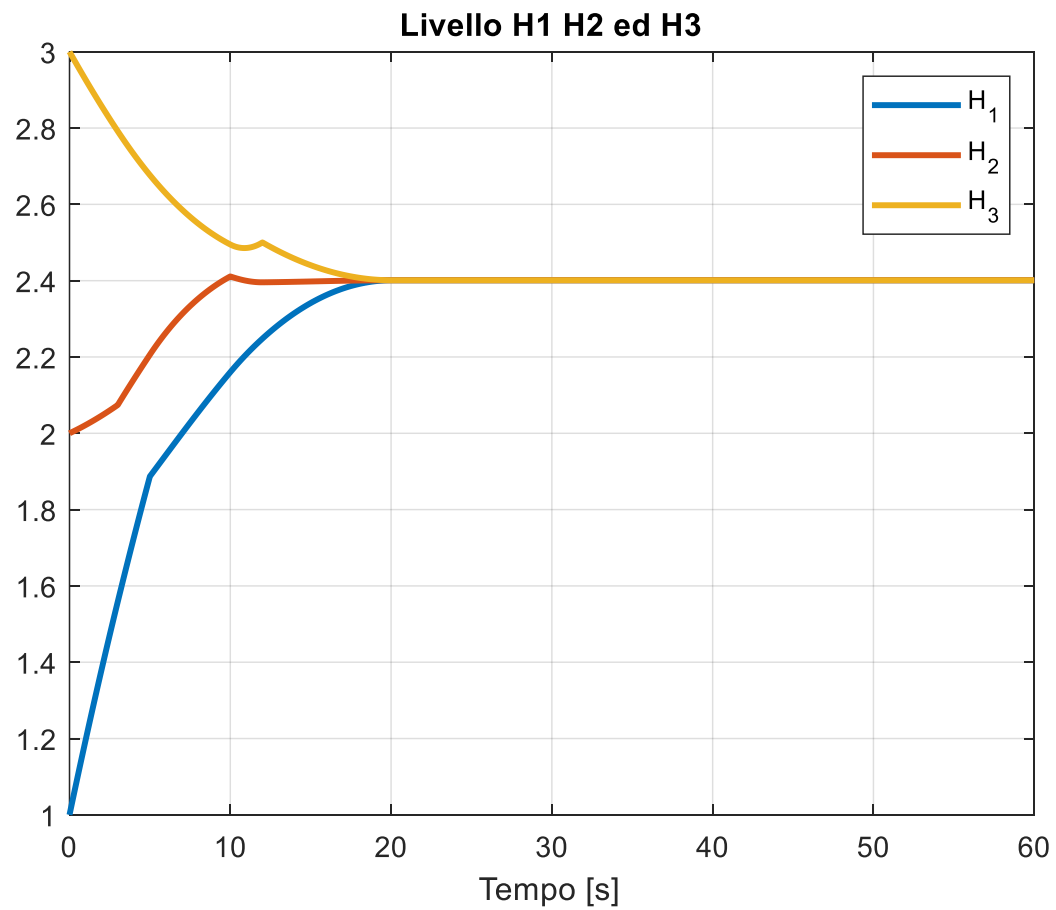
```
plot(H1.Time, [H1.Data H2.Data H3.Data], 'LineWidth', 2)
```

```
grid,
```

```
xlabel('Tempo [s]')
```

```
title('Livello H1 H2 ed H3')
```

```
legend('H_1', 'H_2', 'H_3')
```



Gestione di modelli Simulink attraverso un file Script

Ora desideriamo confrontare fra loro i segnali $H_1(t)$ e $H_2(t)$ ottenuti in corrispondenza di 4 valori distinti del coefficiente di efflusso C_{12}

$$C_{12} = 0.5, 0.8, 1.1, 1.5$$

Realizziamo uno script che gestisca in automatico la riparametrizzazione ed i successivi run del modello Simulink e costruisca i grafici richiesti.

File: `Serbatoi_dati_gestione.m`

```
clear all,clc
```

```
%% Esecuzione dei test di confronto
```

```
% Parametri Test 1
```

```
% Parametri serbatoio #1
```

```
a1=5;
```

```
w1=1;
```

```
% Parametri serbatoio #2
```

```
c2=3;
```

```
w2=1;
```

```
teta2=pi/4;
```

```
% Parametri serbatoio #2
```

```
a3=4;
```

```
w3=2;
```

```
% Coefficienti di efflusso
```

```
C12=0.5;
```

```
C23=0.6;
```

```
% Condizioni iniziali
```

```
H10=1;
```

```
H20=2;
```

```
H30=3;
```

```
sim('Serbatoi_v01')
```

```
H1_test1=H1;
```

```
H2_test1=H2;
```

File: Serbatoi_dati_gestione.m

```
% Coefficiente di efflusso nel Test 2
```

```
C12=0.8;
```

```
sim('Serbatoi_v01')
```

```
H1_test2=H1;
```

```
H2_test2=H2;
```

```
% Coefficiente di efflusso nel Test 3
```

```
C12=1.1;
```

```
sim('Serbatoi_v01')
```

```
H1_test3=H1;
```

```
H2_test3=H2;
```

```
% Coefficiente di efflusso nel Test 4
```

```
C12=1.5;
```

```
sim('Serbatoi_v01')
```

```
H1_test4=H1;
```

```
H2_test4=H2;
```

Ad ogni run del modello, che attiviamo con il comando `sim`, il modello Simulink scrive nel Workspace le variabili H1, H2 ed H3

Dopo ogni run del modello i dati esportati nel workspace devono essere salvati in variabili ausiliarie (in questo caso H1_testx e H2_testx con x=1,2,3,4).

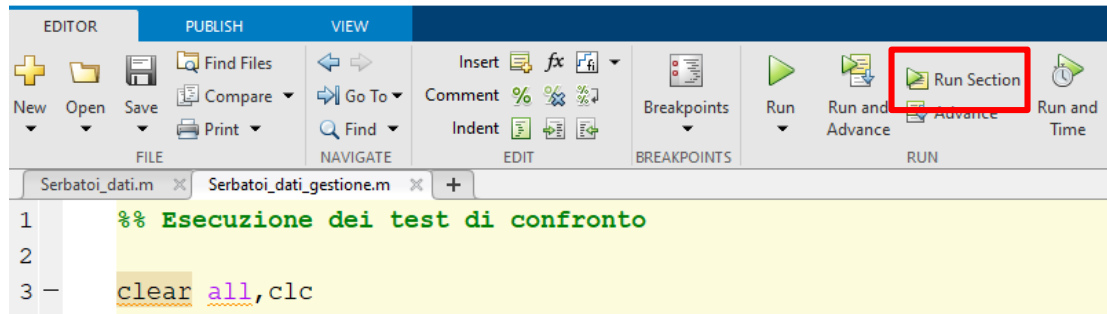
In caso contrario il successivo run del modello sovrascrive i dati della simulazione precedente, che andrebbero persi

Possiamo ora aggiungere allo script una ulteriore Section preposta alla costruzione di grafici di confronto

```
% Creazione grafici
figure(1)
plot(H1_test1.Time, [H1_test1.Data H1_test2.Data H1_test3.Data H1_test4.Data], 'LineWidth', 2)
grid,
xlabel('Tempo [s]')
title('Livelli H1 nei 4 test')
legend('C_{12}=0.5', 'C_{12}=0.8', 'C_{12}=1.1', 'C_{12}=1.5')

figure(2)
plot(H2_test1.Time, [H2_test1.Data H2_test2.Data H2_test3.Data H2_test4.Data], 'LineWidth', 2)
grid,
xlabel('Tempo [s]')
title('Livelli H2 nei 4 test')
legend('C_{12}=0.5', 'C_{12}=0.8', 'C_{12}=1.1', 'C_{12}=1.5')
```

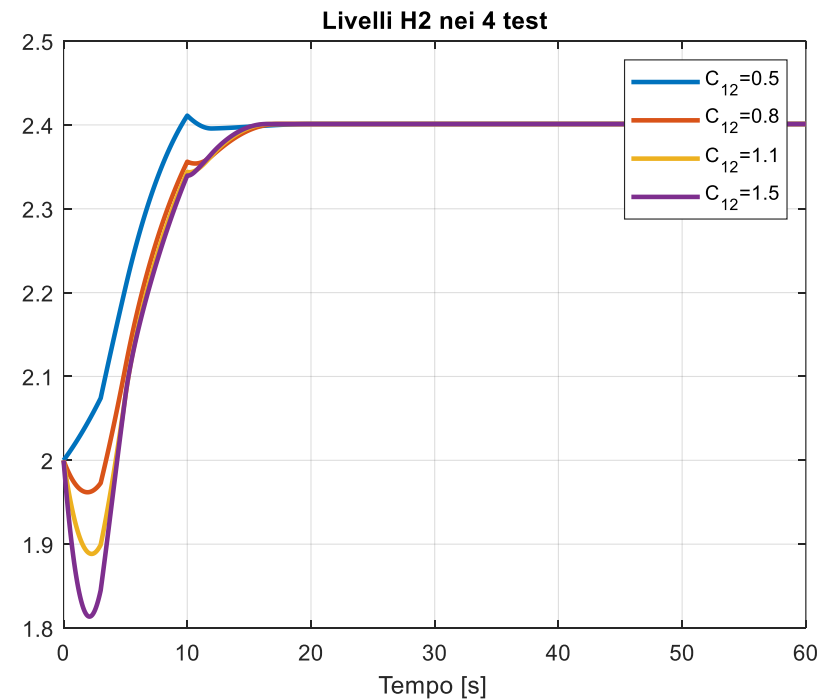
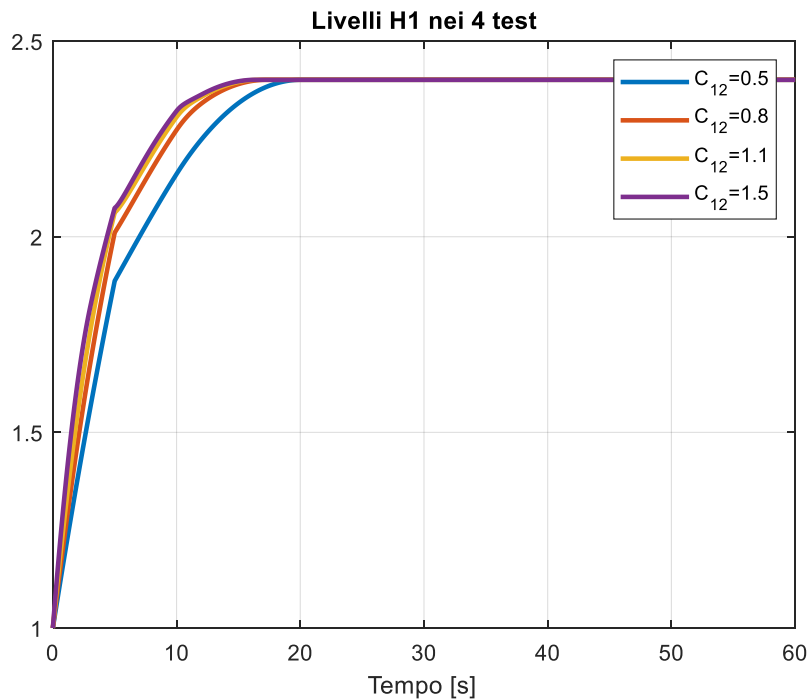
Si esegua la prima Section dello Script (rendere attiva la Section cliccando su di essa, e poi premere «Run Section» o in alternativa Ctrl+Invio)



Workspace		
Name ▲	Value	Size
a1	5	1x1
a3	4	1x1
b2	2	1x1
C12	1.5000	1x1
c2	3	1x1
C23	0.6000	1x1
d2	5	1x1
H1	1x1 double tim...	1x1
H10	1	1x1
H1_test1	1x1 double tim...	1x1
H1_test2	1x1 double tim...	1x1
H1_test3	1x1 double tim...	1x1
H1_test4	1x1 double tim...	1x1
H2	1x1 double tim...	1x1
H20	2	1x1
H2_test1	1x1 double tim...	1x1
H2_test2	1x1 double tim...	1x1
H2_test3	1x1 double tim...	1x1
H2_test4	1x1 double tim...	1x1
H3	1x1 double tim...	1x1
H30	3	1x1
tout	6001x1 double	6001x1
w1	1	1x1
w2	1	1x1
w3	2	1x1

Viene eseguita la sequenza di test, e vengono salvate nel workspace le relative variabili

Eseguendo ora la seconda Section vengono prodotti i seguenti grafici



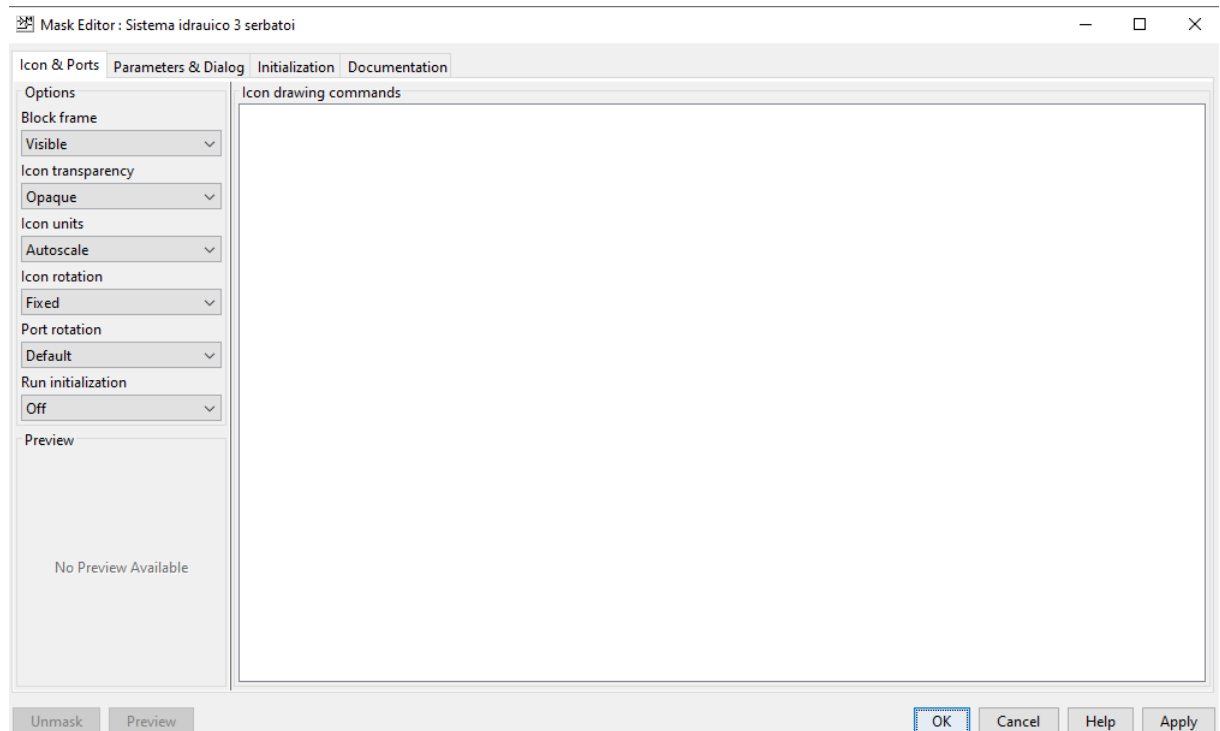
Mascherare (mask) i subsystems

Un Subsystem può essere «**mascherato**». Fra le utilità di tale operazione vi è la possibilità di costruire una comoda **finestra di parametrizzazione** all'interno della quale inserire dei valori per le costanti simboliche utilizzate all'interno del Sottosistema

Partiamo dal modello `Serbatoi_v01.slx` e si clicchi con il tasto destro del mouse sul Subsystem «Sistema idraulico 3 serbatoi» selezionando successivamente dal Menu «Mask» la voce «Create Mask».

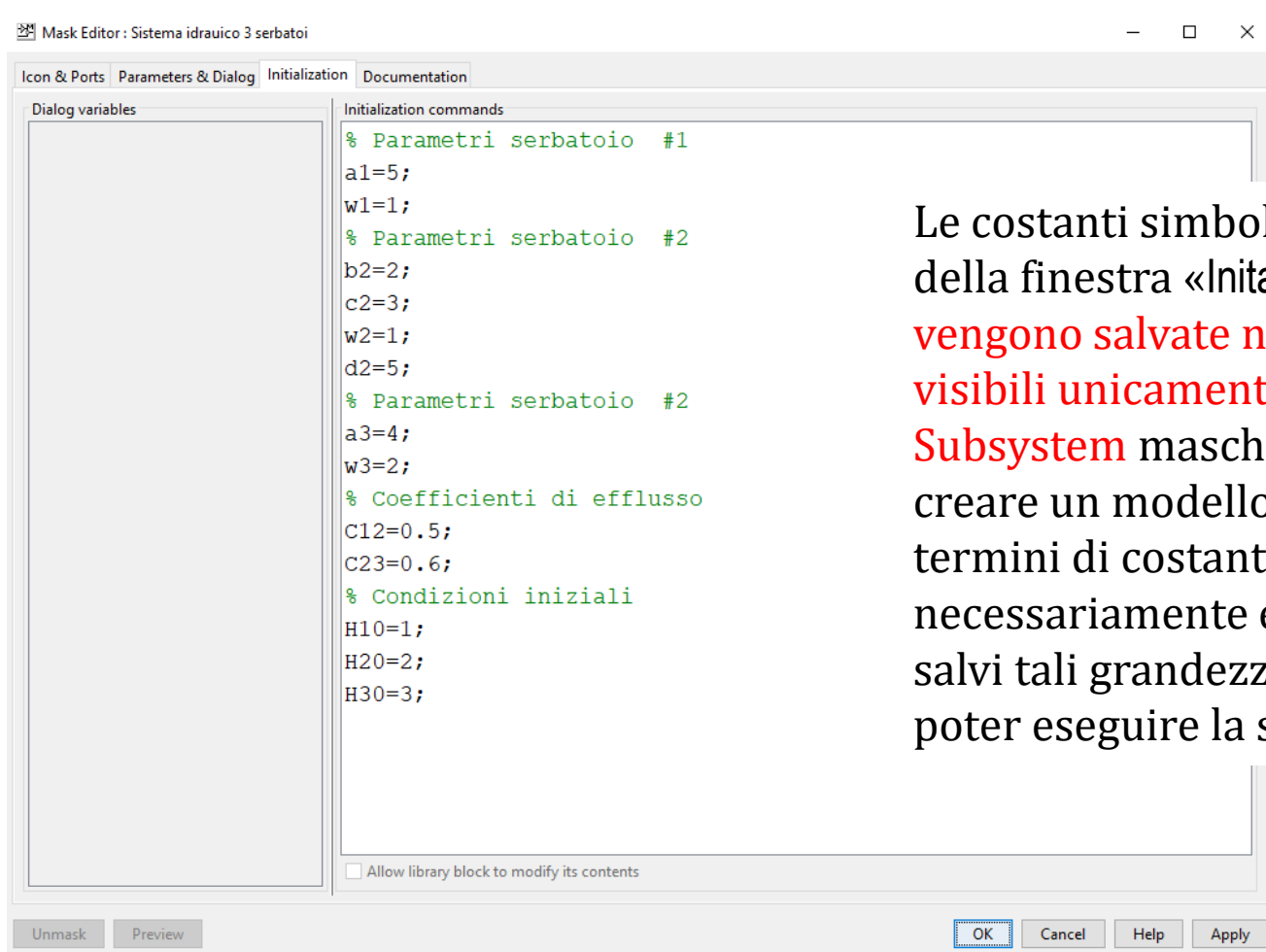
Mask Editor

Vediamone
alcune
funzionalità



Menu “Initialization”

Le istruzioni di assegnazione dei parametri utilizzati all'interno del Subsystem possono essere inserite nella parte di testo, all'interno degli «Initialization commands» anzichè nello script di parametrizzazione.



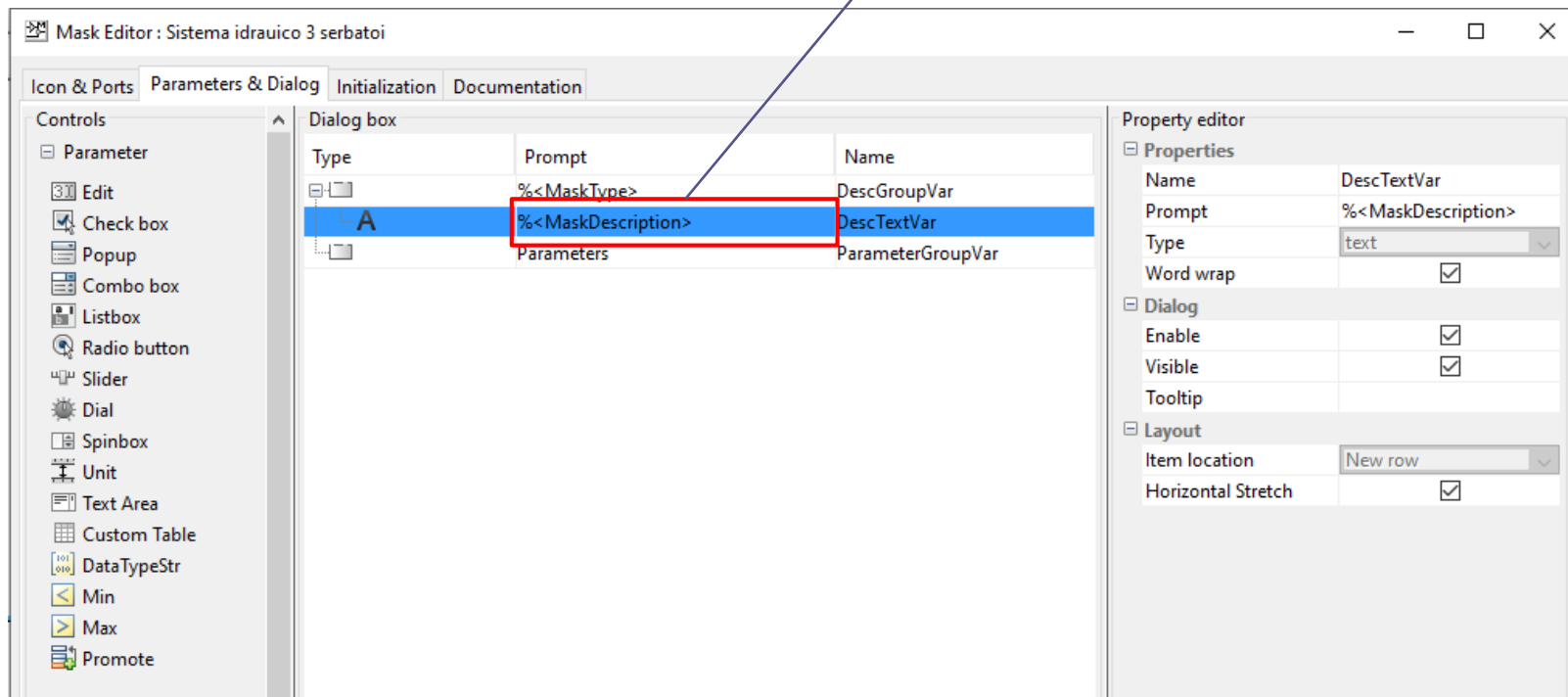
Le costanti simboliche definite all'interno della finestra «Initialization commands» **non vengono salvate nel workspace e sono visibili unicamente all'interno del Subsystem** mascherato. Ciò consente di creare un modello parametrizzato in termini di costanti simboliche senza dover necessariamente eseguire uno script che salvi tali grandezze nel workspace prima di poter eseguire la simulazione.

Vogliamo, in alternativa, creare una **maschera interattiva di parametrizzazione**

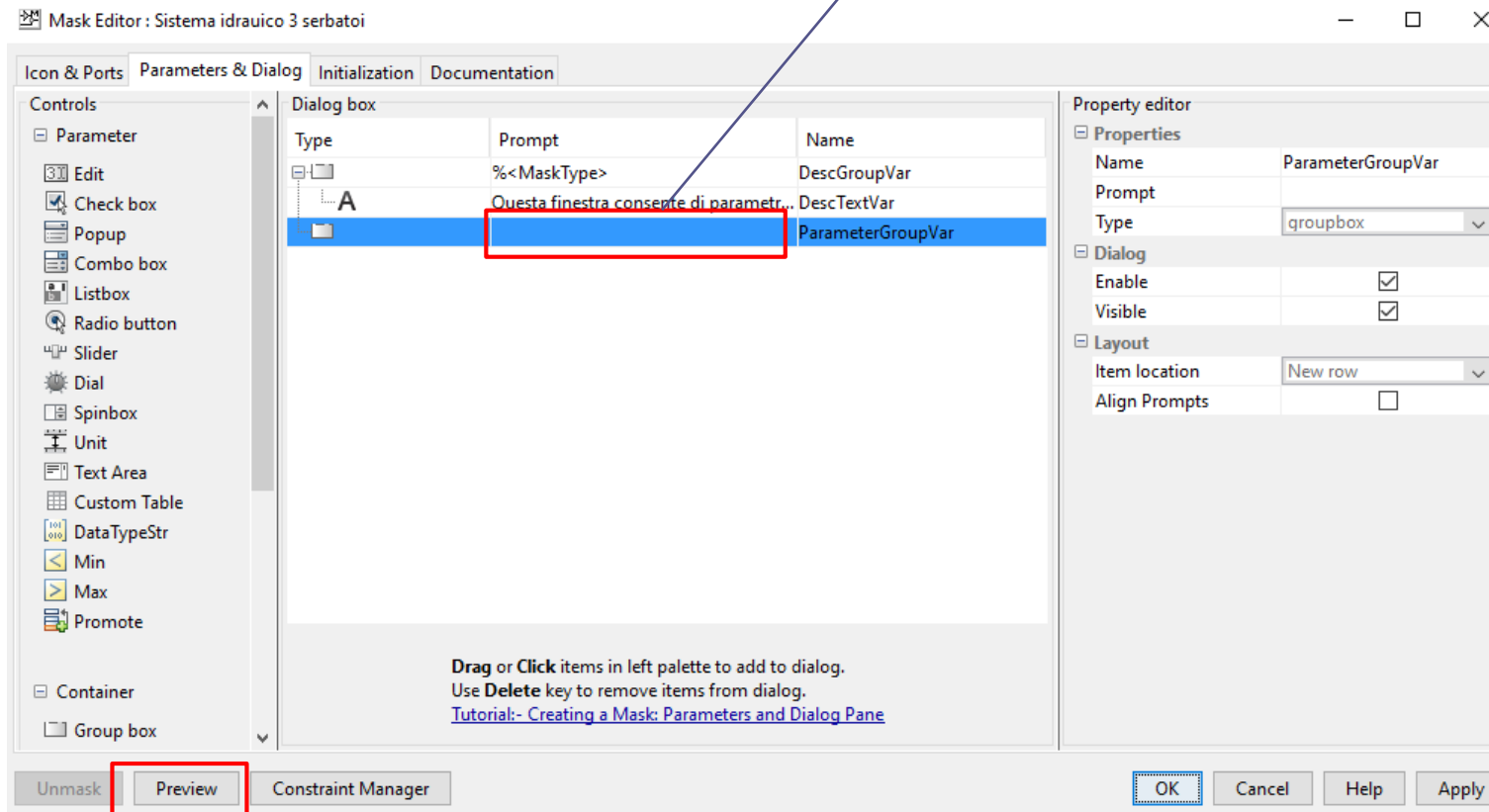
Cancellare le istruzioni dal menu Initialization commands

Selezionare il menu «Parameters & Dialog»

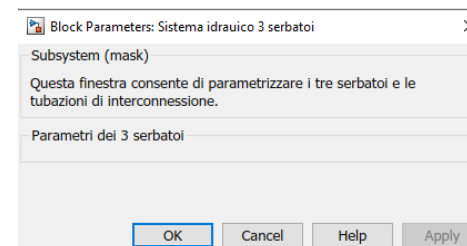
Cliccare su questa casella ed inserire il testo
«*Questa finestra consente di parametrizzare i tre serbatoi e le tubazioni di interconnessione*»



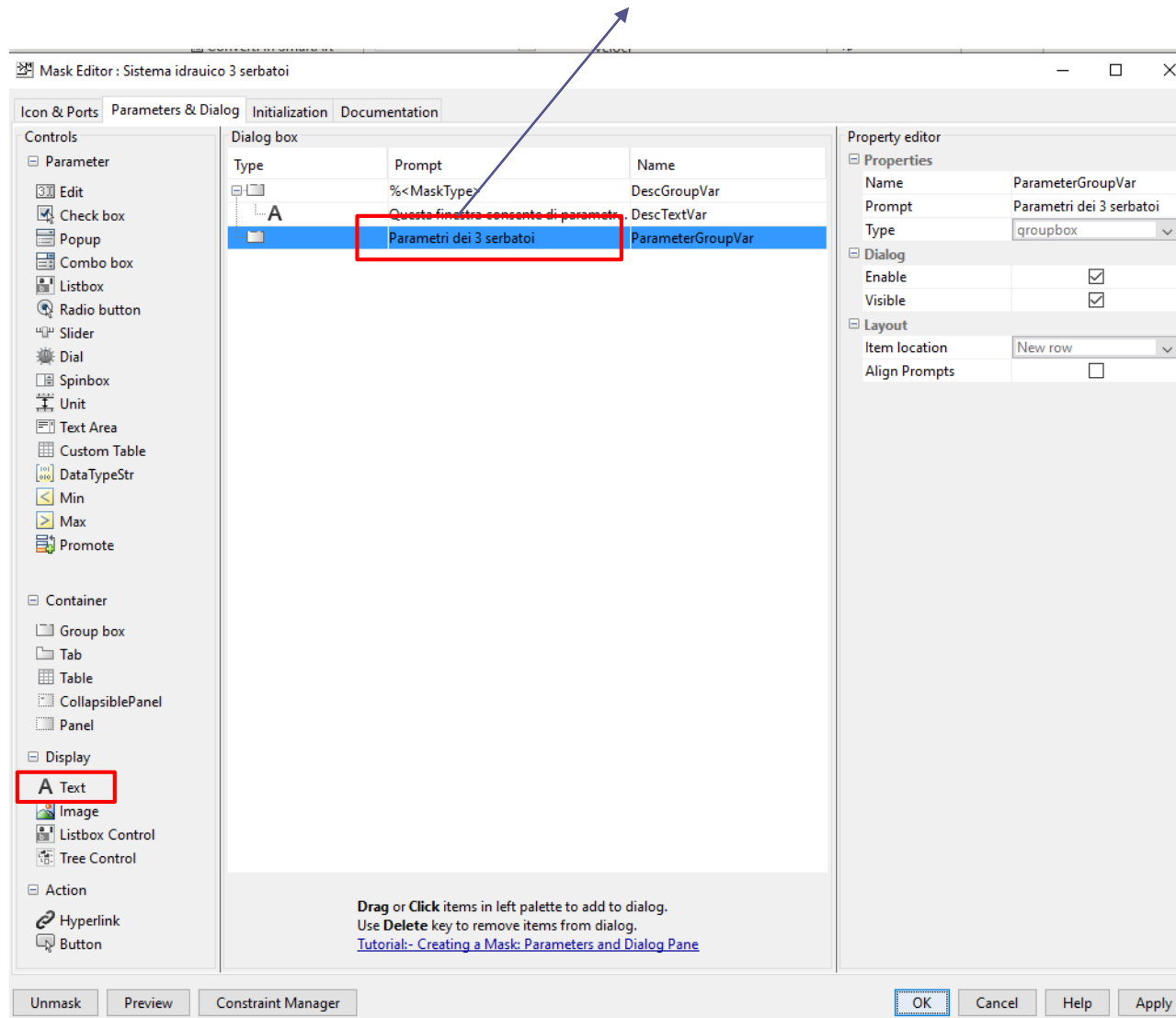
Cliccare su questa casella ed inserire il testo
«Parametri dei 3 serbatoi»



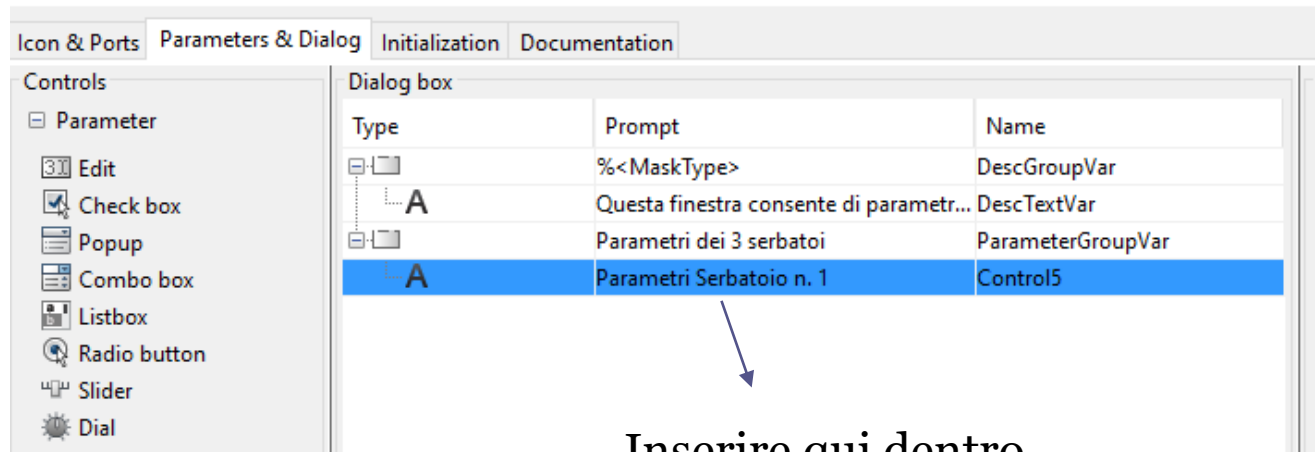
Cliccare successivamente su «Preview»



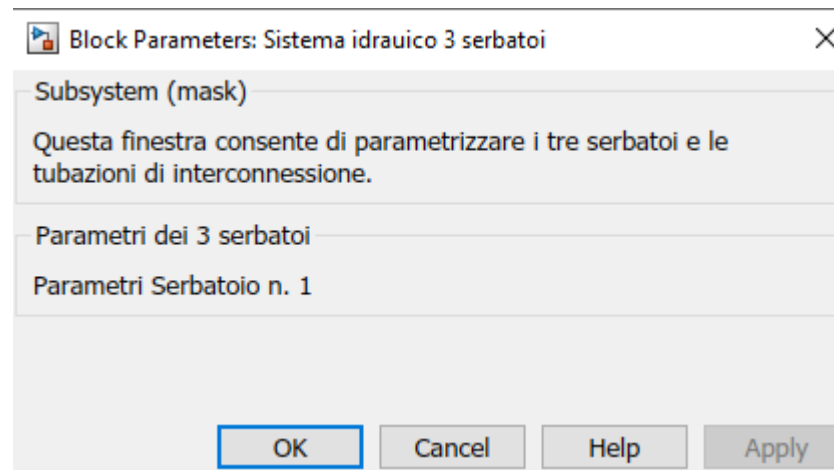
Cliccare su questa casella e successivamente cliccare in basso a sinistra il pulsante Text.



Mask Editor : Sistema idraulico 3 serbatoi



Inserire qui dentro
«*Parametri Serbatoio n.1*»,
e cliccare su «Preview»



Icon & Ports Parameters & Dialog Initialization Documentation

Controls

- Parameter
- Edit**
- Check box
- Popup
- Combo box
- Listbox
- Radio button
- Slider
- Dial
- Spinbox
- Unit
- Text Area
- Custom Table
- DataTypeStr

Dialog box

Type	Prompt	Name
	%<MaskType>	DescGroupVar
A	Questa finestra consente di parametr...	DescTextVar
	Parametri dei 3 serbatoi	ParameterGroupVar
A	Parametri Serbatoio n. 1	Control5

Cliccare su **Edit** nel menu sulla sinistra

Inserire all'interno del campo «Prompt» il testo «*Larghezza (a1) - [m]*» e nel campo «Name» il nome della variabile associata impiegata all'interno del Subsystem mascherato (la variabile **a1**) e cliccare su «Ok».

Icon & Ports Parameters & Dialog Initialization Documentation

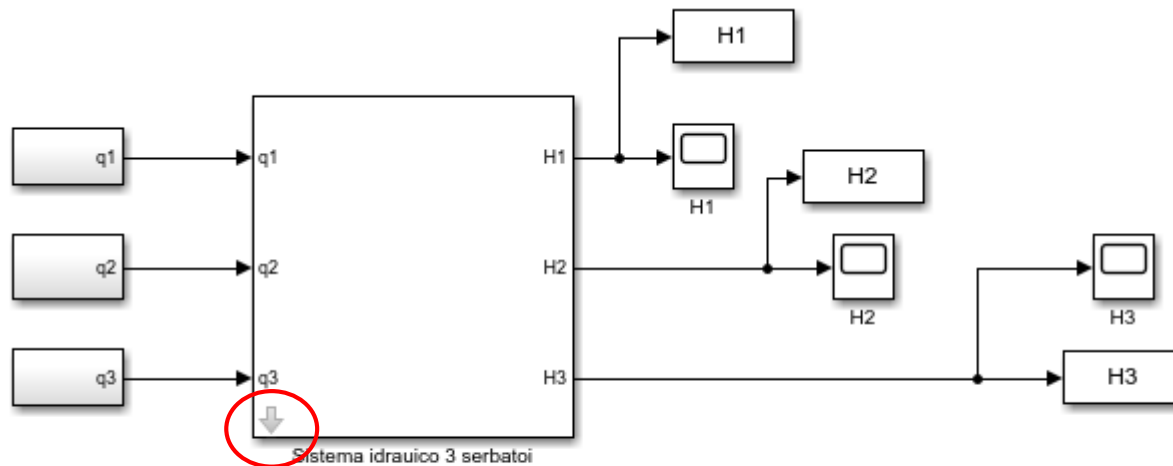
Controls

- Parameter
- Edit**
- Check box
- Popup
- Combo box
- Listbox
- Radio button
- Slider

Dialog box

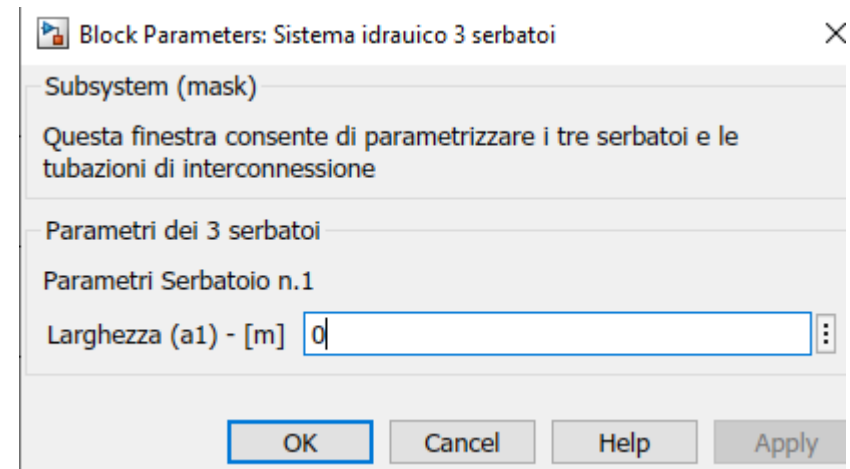
Type	Prompt	Name
	%<MaskType>	DescGroupVar
A	Questa finestra consente di parametr...	DescTextVar
	Parametri dei 3 serbatoi	ParameterGroupVar
A	Parametri Serbatoio n.1	Control2
#1	Larghezza (a1) - [m]	a1

L'aspetto del Subsystem mascherato nella pagina di lavoro è lievemente alterato dalla presenza di una piccola freccetta in basso a sinistra.



Se si fa doppio click sul Subsystem compare la finestra di parametrizzazione, all'interno della quale si può inserire un valore (numerico o simbolico) per la variabile associata alla larghezza del primo serbatoio.

Per accedere al contenuto del Subsystem si deve cliccare sulla freccetta.



Per accedere al menu di configurazione della Mask, cliccare con il tasto destro sul Subsystem e selezionare Mask->Edit Mask (o in alternativa Ctrl+M).

Inserendo nel menu «Parameters & Dialog ulteriori parametri da configurare («**Edit**») – co il relativo testo descrittivo (“Prompt”) e nome della variabile (“Variable”) - ed ulteriori righe testuali («**Text**»), si realizzi la seguente maschera

Block Parameters: Sistema idraulico 3 serbatoi

Subsystem (mask)

Questa finestra consente di parametrizzare i tre serbatoi e le tubazioni di interconnessione

Parametri dei 3 serbatoi

Parametri Serbatoio n.1

Larghezza (a1) - [m] 0

Profondità (w1) - [m] 0

Parametri Serbatoio n.2

Larghezza della base (c2) - [m] 0

Profondità (w2) - [m] 0

Angolo (teta2) - [rad] 0

Parametri Serbatoio n.3

Larghezza (a3) - [m] 0

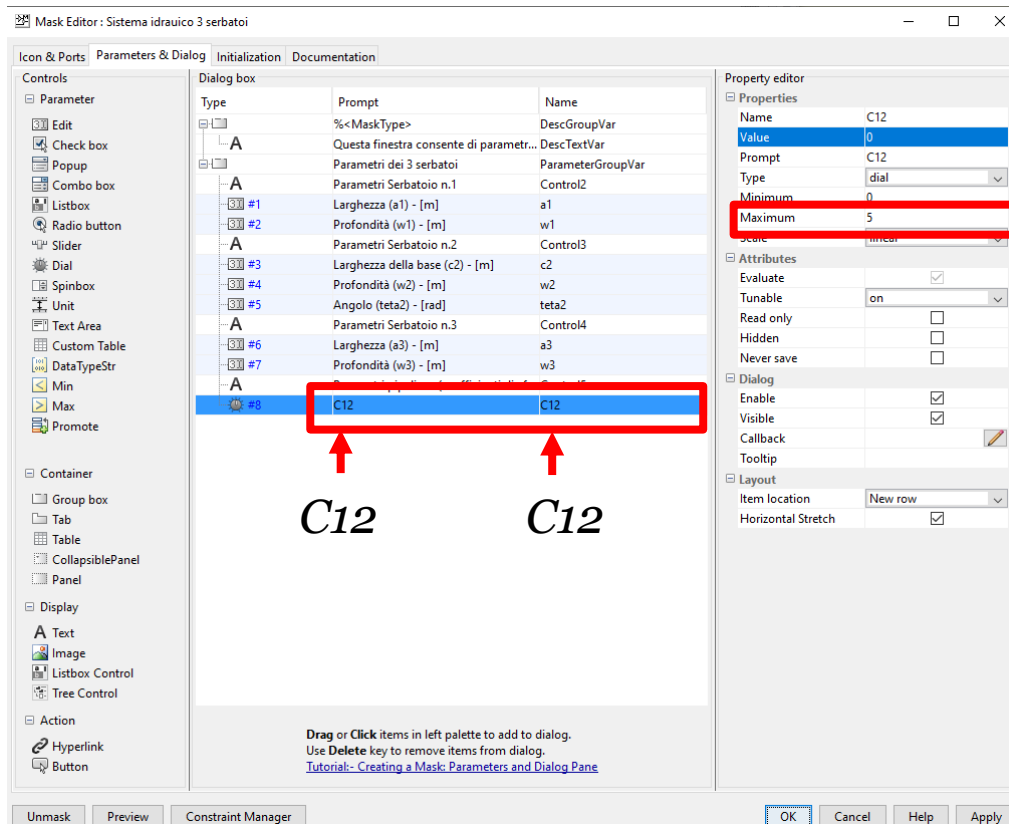
Profondità (w3) - [m] 0

OK Cancel Help Apply

Completiamo la finestra di parametrizzazione con i coefficienti di efflusso.
 Impieghiamo una modalità differente di inserimento del parametro.

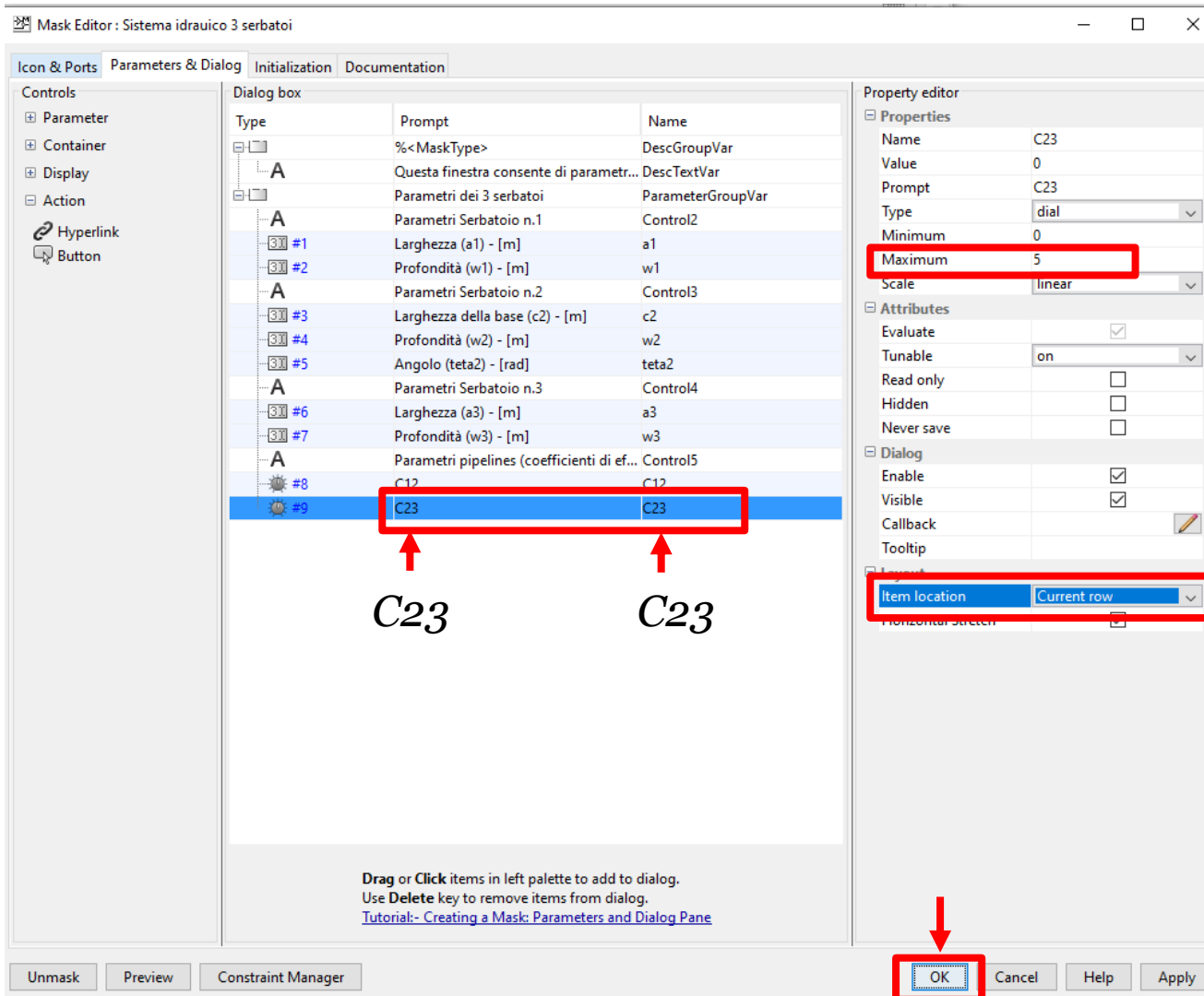
Inseriamo preliminarmente una riga di testo «*Parametri pipelines (coefficienti di efflusso in $m^{(5/2)}/s$)*»

Successivamente, selezioniamo dal menu sulla sinistra la voce «Dial» (anziché «Edit»).



Maximum = 5

Selezioniamo nuovamente «Dial»



Maximum = 10

*Item location =
Current Row*

La finestra di parametrizzazione viene completata come segue.

Block Parameters: Sistema idraulico 3 serbatoi

Subsystem (mask)

Questa finestra consente di parametrizzare i tre serbatoi e le tubazioni di interconnessione

Parametri dei 3 serbatoi

Parametri Serbatoio n.1

Larghezza (a1) - [m] 0

Profondità (w1) - [m] 0

Parametri Serbatoio n.2

Larghezza della base (c2) - [m] 0

Profondità (w2) - [m] 0

Angolo (teta2) - [rad] 0

Parametri Serbatoio n.3

Larghezza (a3) - [m] 0

Profondità (w3) - [m] 0

Parametri pipelines (coefficienti di efflusso in $m^{5/2}/s$)

C12

0.0 5.0

1.850

C23

0.0 10.0

0

OK Cancel Help Apply

I coefficienti di efflusso possono essere parametrizzati attraverso delle **manopole** (ma anche inserendo il valore nella casella di testo sottostante)

Block Parameters: Sistema idraulico 3 serbatoi

Subsystem (mask)

Questa finestra consente di parametrizzare i tre serbatoi e le tubazioni di interconnessione

Parametri dei 3 serbatoi

Parametri Serbatoio n.1

Larghezza (a1) - [m]

Profondita (w1) - [m]

Parametri Serbatoio n.2

Larghezza della base (c2) - [m]

Profondita (w2) - [m]

Angolo (teta2) - [rad]

Parametri Serbatoio n.3

Larghezza (a3) - [m]

Profondita (w3) - [m]

Parametri pipelines (coefficienti di efflusso in $m^{5/2}/s$)

C12

C23

Condizioni iniziali

H1(0) [m] H2(0) [m] H3(0) [m]

OK Cancel Help Apply

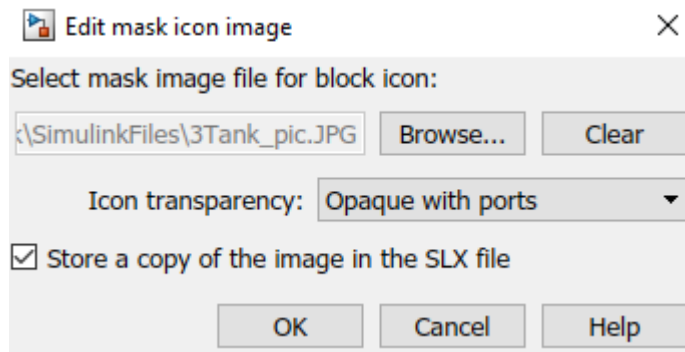
Inseriamo (mediante il comando «Edit» nel menu a sinistra) tre ulteriori caselle testuali in cui andare ad inserire i valori delle condizioni iniziali, disponendole una affianco all'altra.

La versione finale della finestra di parametrizzazione è mostrata a lato.

Ora il modello può essere eseguito inserendo i valori numerici dei parametri (o se opportuno delle costanti simboliche) nelle rispettive caselle.

Concludiamo mostrando come inserire all'interno di un blocco Subsystem mascherato un file immagine volto a rendere maggiormente efficace la rappresentazione del modello di simulazione.

Si clicchi con il tasto destro sul Subsystem, e si selezioni Mask->Add Icon Image

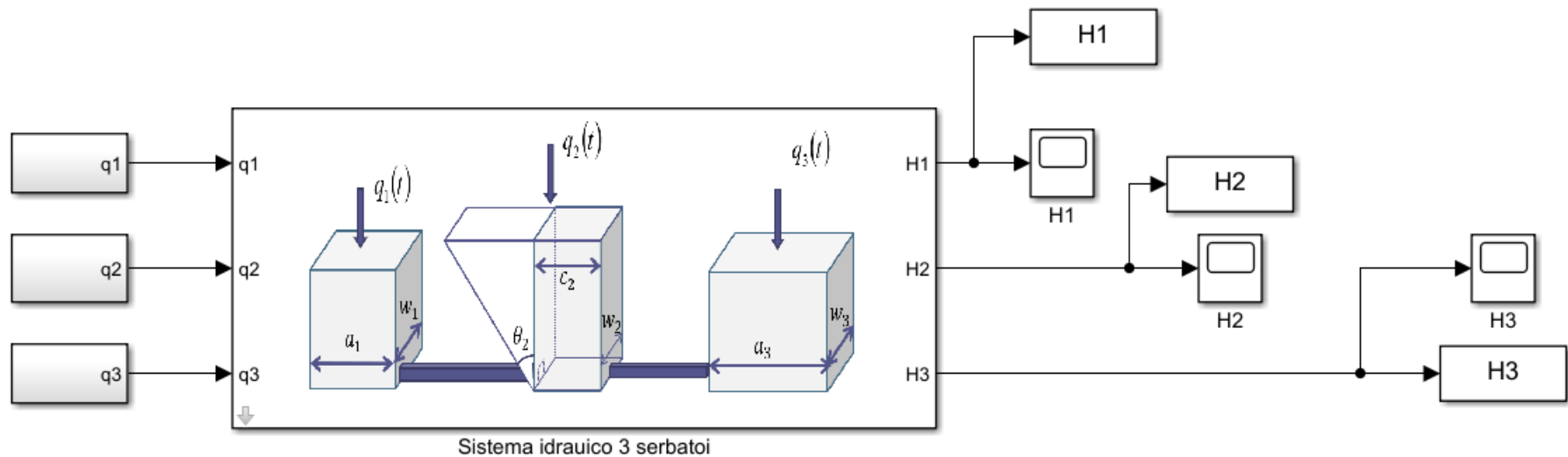


Selezionare il file «3Tank_pic.jpg» dalla cartella di files fornita.

L'opzione «Opaque with ports» fa sì che nella visualizzazione del Subsystem vengano mostrati i nomi delle porte di input ed output.

L'opzione «Store a copy....» salva l'immagine all'interno del file slx, in modo che anche se il file immagine sia cancellato o spostato esso continui ad essere visualizzato all'interno del modello.

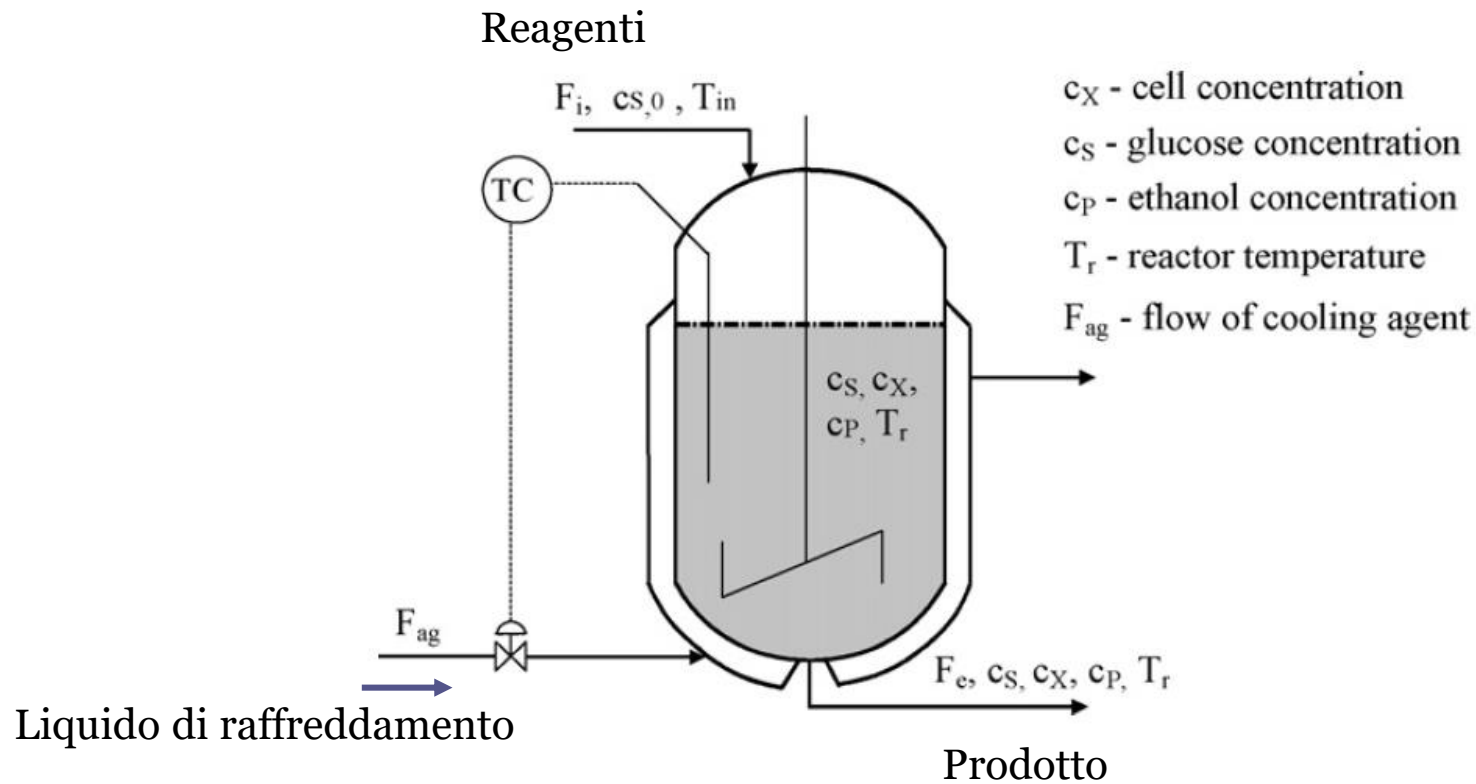
Aspetto finale del modello



File:

Serbatoi_v01MASK.slx

Modellazione dinamica di un processo di fermentazione



Modelliamo il funzionamento di un bioreattore all'interno del quale avviene un processo di fermentazione.

Bioreattore industriale.



Variabili di processo

$C_X(t)$ Concentrazione della biomassa $[g/\ell]$

$C_P(t)$ Concentrazione dell'etanolo nel prodotto $[g/\ell]$

$C_S(t)$ Concentrazione del glucosio nel substrato $[g/\ell]$

$C_{O_2}(t)$ Concentrazione dell'ossigeno disciolto $[g/\ell]$

$T_r(t)$ Temperatura all'interno del reattore $[^{\circ}C]$

$T_{ag}(t)$ Temperatura del liquido di raffreddamento $[^{\circ}C]$

Ingressi tempo-varianti

$F_{ag}(t)$ Portata del liquido di raffreddamento $[\ell/h]$

Grandezze costanti

F_i	Portata del substrato in ingresso al reattore [ℓ/h]
F_e	Portata del substrato all'uscita del reattore [ℓ/h]
T_{in}	Temperatura del substrato in ingresso al reattore [$^{\circ}C$]
$T_{in,ag}$	Temperatura del liquido di raffreddamento all'ingresso del jacket [$^{\circ}C$]
$C_{S,in}$	Concentrazione del substrato in ingresso al reattore [g/ℓ]

Modello matematico

$$\frac{dC_X(t)}{dt} = \mu_X(T_r) C_X(t) \frac{C_S(t)}{K_S + C_S(t)} e^{-K_P C_P(t)} - \frac{F_e(t)}{V} C_X(t)$$

$$\frac{dC_P(t)}{dt} = \mu_P C_X(t) \frac{C_S(t)}{K_{S1} + C_S(t)} e^{-K_{P1} C_P(t)} - \frac{F_e(t)}{V} C_P(t)$$

$$\begin{aligned} \frac{dC_S(t)}{dt} = & -\frac{1}{R_{SX}} \mu_X(T_r) C_X(t) \frac{C_S(t)}{K_S + C_S(t)} e^{-K_P C_P(t)} - \frac{1}{R_{SP}} \mu_P C_X(t) \frac{C_S(t)}{K_{S1} + C_S(t)} e^{-K_{P1} C_P(t)} \\ & + \frac{F_i}{V} C_{S,in} - \frac{F_e}{V} C_S(t) \end{aligned}$$

$$\frac{dC_{O_2}(t)}{dt} = k_{la}(T_r) (C_{O_2,eq}(T_r) - C_{O_2}(t)) - r_{O_2}(C_{O_2}, C_X) - \frac{F_e}{V} C_{O_2}(t)$$

$$\frac{dT_r(t)}{dt} = \frac{F_i}{V} (T_{in}(t) + 273) - \frac{F_e}{V} (T_r(t) + 273) + \frac{r_{O_2}(C_{O_2}, C_X) \Delta H_r}{32 \rho_r C_{heat,r}} - \frac{K_T A_T (T_r(t) - T_{ag}(t))}{V \rho_r C_{heat,r}}$$

$$\frac{dT_{ag}(t)}{dt} = \frac{F_{ag}(t)}{V_j} (T_{in,ag} - T_{ag}(t)) + \frac{K_T A_T (T_r(t) - T_{ag}(t))}{V_j \rho_{ag} C_{heat,ag}}$$

Parametri dipendenti dalle variabili di processo

$$\mu_X(T_r) = A_1 e^{-\frac{E_{a1}}{R(T_r(t)+273)}} - A_2 e^{-\frac{E_{a2}}{R(T_r(t)+273)}}$$

$$r_{O_2}(C_{O_2}, C_X) = 1000 \mu_{O_2} \frac{C_{O_2}(t) C_X(t)}{Y_{O_2} + C_{O_2}(t)}$$

$$k_{la}(T_r) = k_{\ell a,0} 1.024^{(T_r(t)-20)}$$

$$C_{O_2,eq}(T_r) = [14.16 - 0.3943 T_r(t) + 0.007714 T_r^2(t) - 0.0000646 T_r^3(t)] 10^{-\Sigma_{HT}}$$

Costanti fisiche presenti nel modello

Costanti cinetiche

$$A_1 = 9.5 \cdot 10^8$$

$$A_2 = 2.55 \cdot 10^{33}$$

$$E_{a1} = 55000 \text{ J/mol}$$

$$E_{a2} = 220000 \text{ J/mol}$$

$$R = 8.31 \text{ J/(mol } ^\circ\text{K)}$$

$$\mu_P = 1.79 \text{ } \ell/h$$

$$K_S = 1.03 \text{ g/}\ell$$

$$K_P = 0.139 \text{ g/}\ell$$

$$K_{S1} = 1.68 \text{ g/}\ell$$

$$K_{P1} = 0.07 \text{ g/}\ell$$

$$R_{SX} = 0.607$$

$$R_{SP} = 0.435$$

$$Y_{O2} = 0.97 \text{ mg/mg}$$

$$K_{O2} = 8.86 \text{ mg/}\ell$$

$$\mu_{O2} = 0.5 \text{ } \ell/h$$

Costanti termodinamiche

$$k_{\ell a,0} = 38 \text{ } \ell/h$$

$$K_T = 3.6 \cdot 10^5 \text{ J/(h m}^2 \text{ } ^\circ\text{K)}$$

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

$$\rho_r = 1080 \text{ g/}\ell$$

$$\rho_{ag} = 1000 \text{ g/}\ell$$

$$C_{heat,r} = 4.18 \text{ J/(g } ^\circ\text{K)}$$

$$C_{heat,ag} = 4.18 \text{ J/(g } ^\circ\text{K)}$$

$$\Delta H_r = 518 \text{ kJ / mol}$$

Costanti fisiche presenti nel modello

Costanti ioniche specifiche [ℓ/g ion]

$$H_{Na} = -0.55$$

$$H_{Ca} = -0.303$$

$$H_{Mg} = -0.314$$

$$H_H = -0.774$$

$$H_{Cl} = 0.844$$

$$H_{CO_3} = 0.485$$

$$H_{HO} = 0.941$$

Pesi molecolari [g / rmol]

$$M_{NaCl} = 58.5$$

$$M_{CaCO_3} = 90$$

$$M_{MgCl_2} = 95$$

$$M_{Na} = 23$$

$$M_{Ca} = 40$$

$$M_{Mg} = 24$$

$$M_{Cl} = 35.5$$

$$M_{CO_3} = 60$$

Costanti fisiche presenti nel modello

Dati del bioreattore

$$m_{NaCl} = 500 \text{ g} \quad F_i = 51 \text{ } \ell/h$$

$$m_{CaCO_3} = 100 \text{ g} \quad F_e = 51 \text{ } \ell/h$$

$$m_{MgCl_2} = 100 \text{ g}$$

$$F_{ag,nom} = 18 \text{ } \ell/h \quad \text{Ingresso costante nominale}$$

(portata del liquido di raffreddamento)

$$V_j = 50 \text{ } \ell \quad T_{in} = 25 \text{ } ^\circ C$$

$$V = 1000 \text{ } \ell \quad T_{in,ag} = 15 \text{ } ^\circ C$$

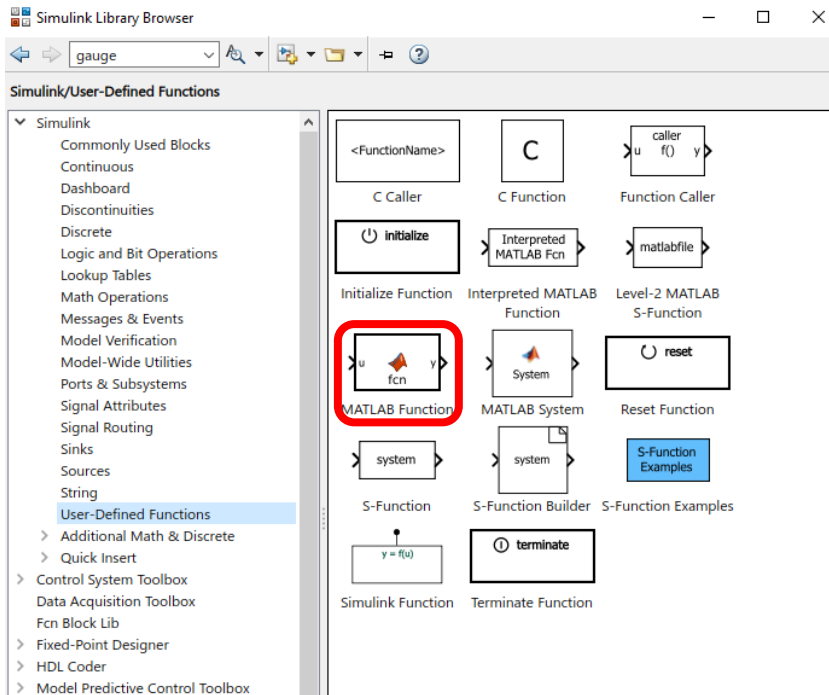
$$pH = 6 \quad C_{S,in} = 60 \text{ g/}\ell$$

$$\begin{aligned} \Sigma_{HT} = & \frac{0.5H_{Na} m_{NaCl} M_{Na}}{M_{NaCl} V} + \frac{2H_{Ca} m_{CaCO_3} M_{Ca}}{M_{CaCO_3} V} + \frac{2H_{Mg} m_{MgCl_2} M_{Mg}}{M_{MgCl_2} V} \\ & + 0.5H_{Cl} \left(\frac{m_{NaCl}}{M_{NaCl}} + \frac{2m_{MgCl_2}}{M_{MgCl_2}} \right) \frac{M_{Cl}}{V} + \frac{2H_{CO_3} m_{CaCO_3} M_{CO_3}}{M_{CaCO_3} V} \\ & + 0.5 H_H 10^{-pH} + 0.5 H_{HO} 10^{-(14-pH)} \end{aligned}$$

La modellazione dinamica del bioreattore è non banale per la relativa complessità del modello matematico.

Illustriamo un approccio alla modellazione in Simulink che consente di gestire modelli matematici anche molto complessi senza che la complessità della rappresentazione tramite schema a blocchi «esploda».

Faremo uso del blocco « Matlab Function» (libreria: User-defined functions)

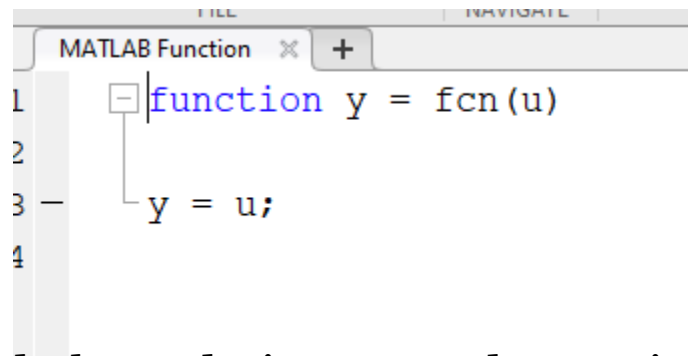


Consente di scrivere ed eseguire un “function file” Matlab direttamente all’interno di un modello Simulink.

Default



Se si fa doppio click sul blocco lo si apre nell'editor testuale



```
MATLAB Function  x +  
1  function y = fcn(u)  
2  
3  y = u;  
4
```

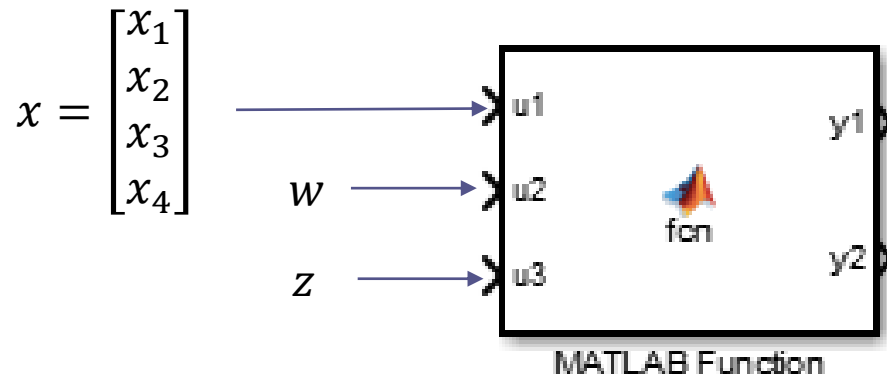
Il blocco si programma secondo le medesime procedure e sintassi viste a suo tempo per la redazione di un Function File. Può essere configurato con un numero arbitrario di porte di ingresso e uscita, in relazione al numero di ingressi e uscite della function. Ciascun ingresso o uscita può essere un segnale scalare, vettoriale o matriciale.

Si possono eseguire complesse operazioni matriciali che rendono particolarmente utile questo blocco elementare nell'ambito di modelli di particolare complessità.

All'interno di un blocco Matlab function **non sono visibili le variabili del workspace**, neanche se queste vengono definite come globali.

Sviluppiamo un esempio preliminare

Vogliamo realizzare un blocco che riceva 3 quantità in ingresso, la prima delle quali è un vettore che contiene 4 segnali, mentre i restanti due ingressi sono quantità scalari (v. Figura)



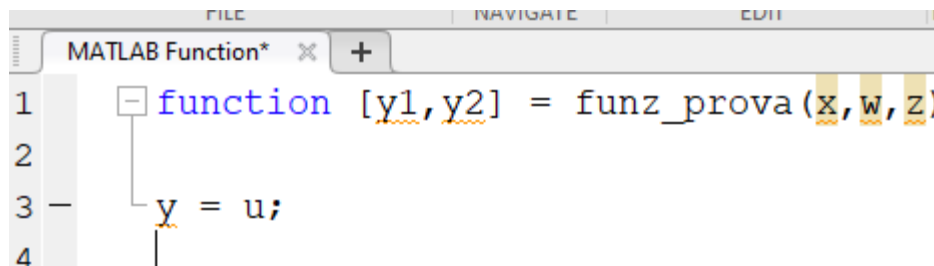
Il blocco deve restituire all'esterno le due quantità y_1 ed y_2 definite come segue

$$y_1 = x_1 + x_2 + x_3 + x_4 + w + z$$

$$y_2 = [x_1 + x_2 \quad x_3 + x_4 \quad w \quad z]$$

Modifichiamo il contenuto del blocco all'interno dell'editor.

Il primo passo è modificare la prima riga, all'interno della quale si impostano il numero di ingressi ed uscite, il nome che sarà associato ai vari terminali di ingresso e uscita del blocco ed il nome della funzione



```

1 function [y1,y2] = funz_prova(x,w,z)
2
3 y = u;
4

```

L'aspetto del blocco cambia. Compaiono i nuovi terminali di input ed output ed il nome della funzione



Secondo le sintassi viste a suo tempo per la redazione di una Function, completiamo il codice come segue

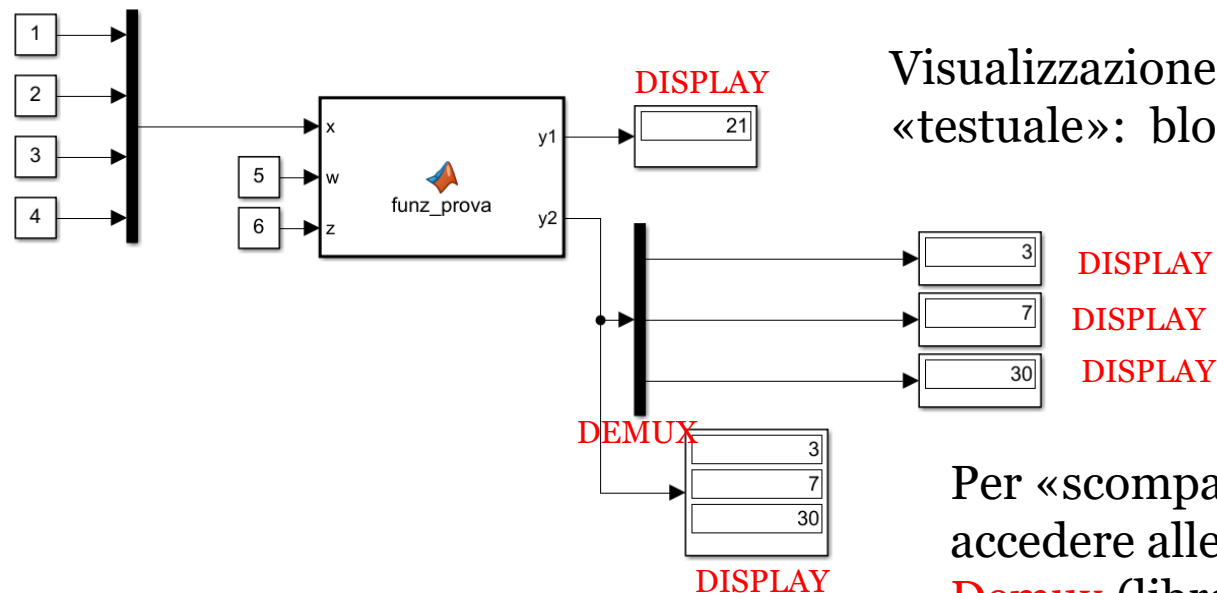
```
function [y1,y2] = funz_prova(x,w,z)
```

```
y1=sum(x)+w+z;
```

```
% Sintassi alternativa:
```

```
% y1=x(1)+x(2)+x(3)+x(4)+w+z;
```

```
y2=[x(1)+x(2)    x(3)+x(4)    w*z];
```

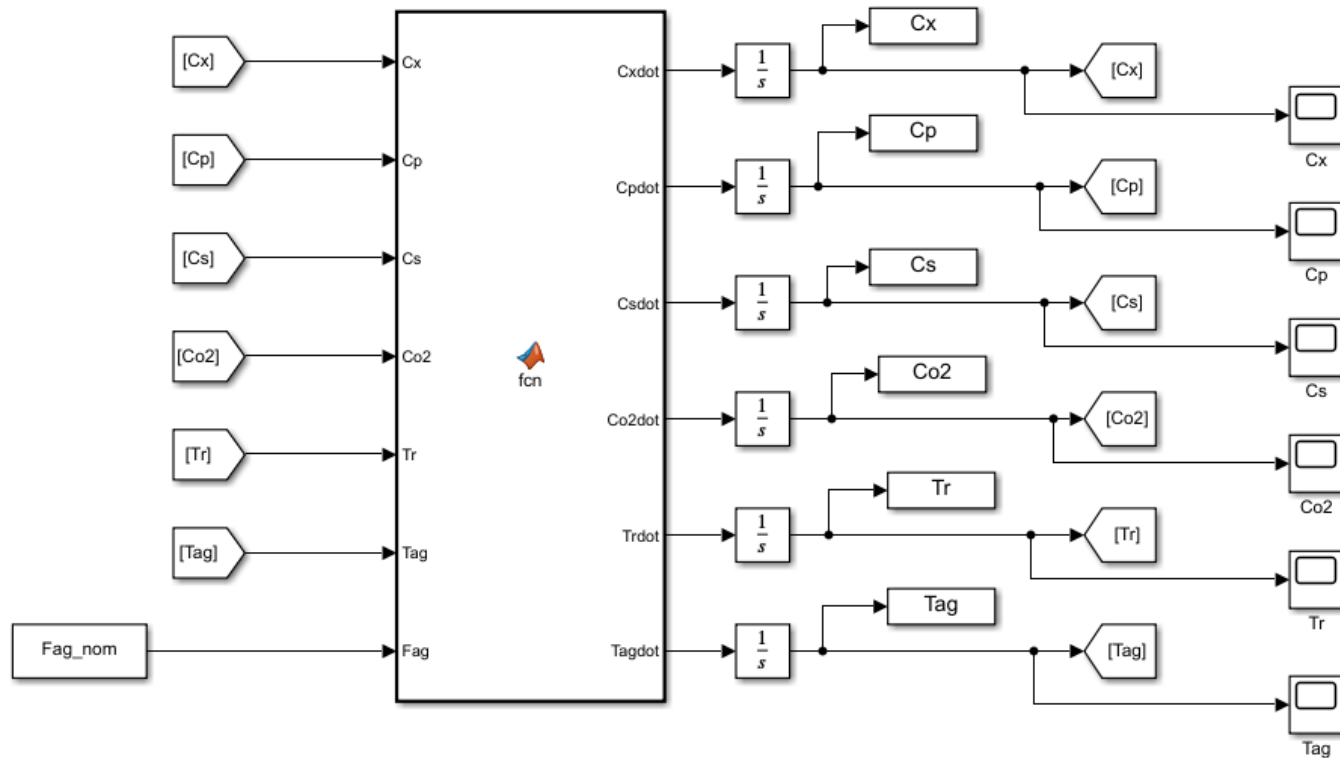


Visualizzazione di segnali in formato «testuale»: blocco **display** (libreria: Sinks)

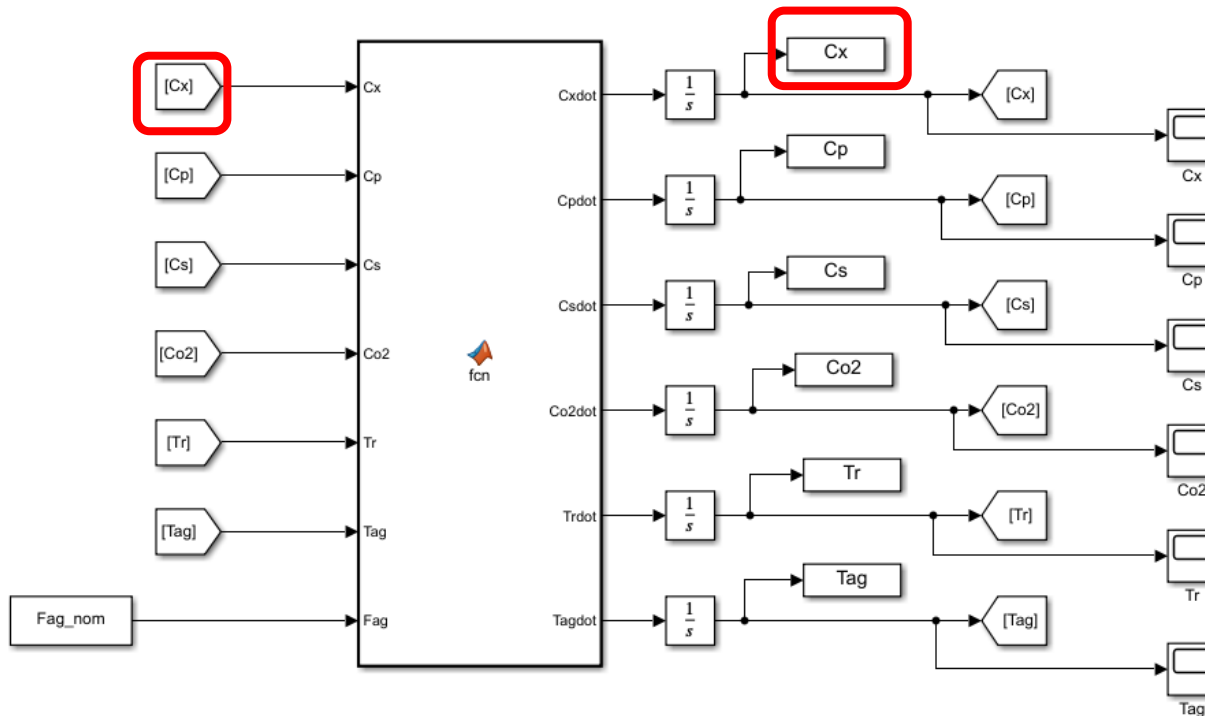
Per «scompattare» un segnale vettoriale ed accedere alle singole componenti: blocco **Demux** (libreria: Commonly Used Blocks)

Iniziamo lo sviluppo del modello del bioreattore.

La struttura generale scelta è la seguente: sei integratori, posti in parallelo, genereranno ai loro terminali di uscita le sei variabili di processo ed un blocco Matlab Function costruirà i segnali di ingresso ai vari blocchi integratori (le derivate temporali delle variabili di processo, la cui espressione è data dalle sei equazioni differenziali accoppiate in forma esplicita)



Per realizzare il modello del bioreattore impiegheremo anche due blocchi elementari non ancora visti, denominati «Goto» e «From» (Libreria «Signal Routing»), evidenziati in rosso nell'immagine in basso, che servono per portare un segnale da un punto ad un altro di un modello Simulink senza necessita di tracciare una linea di connessione (una sorta di trasmissione «wireless» del segnale)



Si crei e si esegua il seguente Script per memorizzare nel workspace 7 variabili simboliche che contengono i valori delle 6 variabili di processo all'istante iniziale ed una ulteriore costante utile per la costruzione del modello.

```
clear all
close all
clc
```

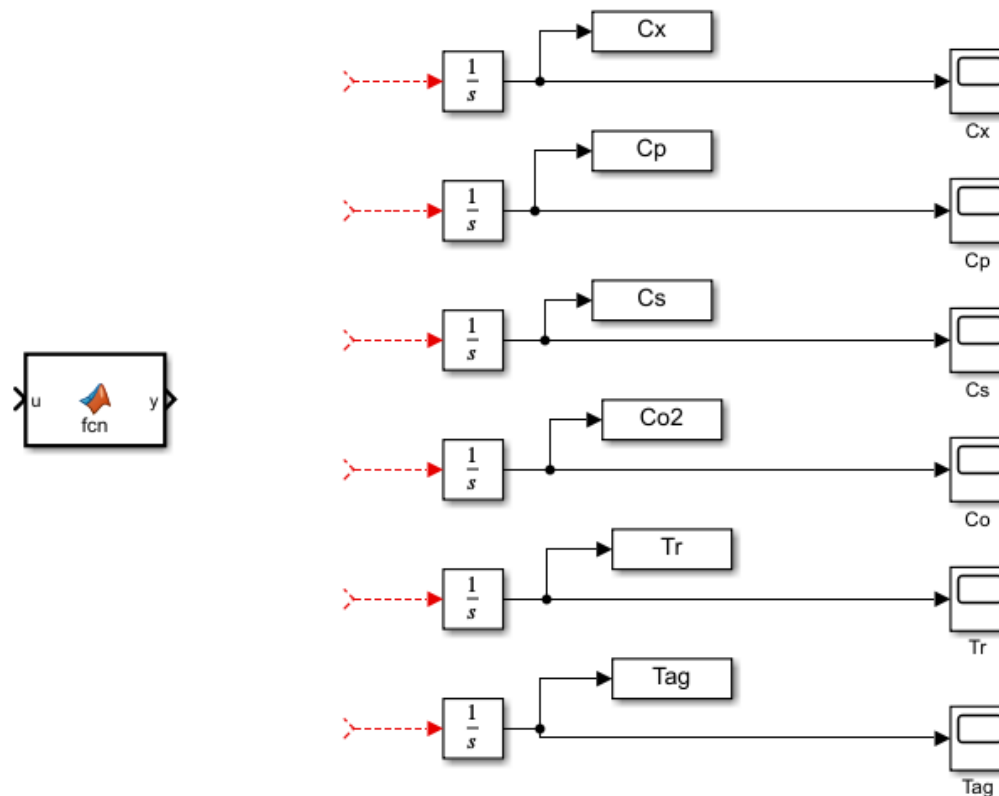
```
Fag_nom=18; %portata costante del
            %liquido di raffreddamento [l/h]
```

```
% CONDIZIONI INIZIALI
```

```
Cx0=1; %biomass concentration [g/l]
Cp0=13; %Product/ethanol concentration [g/l]
Cs0=27; %Substrate concentration [g/l]
Co20=2.5; %Oxygen concentration [mg/l]
Tr0=26; %Reactor temperature [°C]
Tag0=29; %Cooling agent temperature [°C]
```

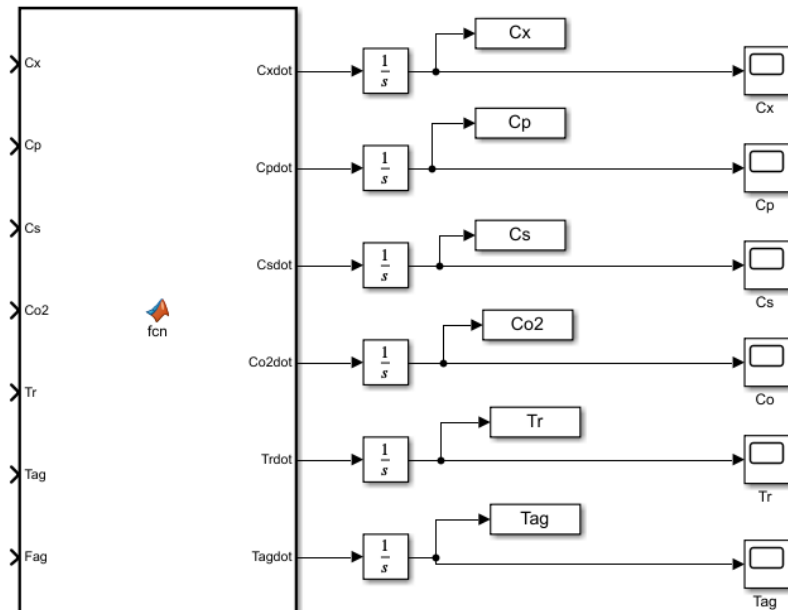
File: Bioreattore_dati.m

Importiamo nella pagina di lavoro i sei integratori, configuriamone le condizioni iniziali, e colleghiamo all'uscita di ogni integratore un blocco Scope ed un blocco «To workspace» per esportare i dati generati dalla simulazione nel workspace di Matlab. Si collochi quindi alla sinistra un blocco Matlab function.



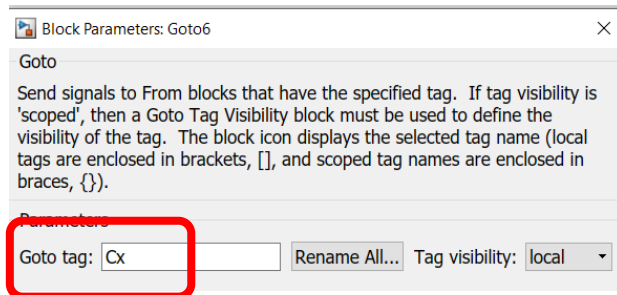
Dobbiamo configurare il blocco Matlab Function in modo che restituisca in uscita 6 segnali (le derivate temporali delle variabili di processo). Il blocco dovrà ricevere in ingresso tutti quei segnali necessari per il calcolo delle derivate temporali delle variabili di processo. L'ispezione del sistema di equazioni differenziali che governa il processo rivela come dovranno essere passate in ingresso al blocco Matlab Function le sei variabili di progresso più l'ingresso tempo-variante $F_{ag}(t)$ (che rappresenta la portata del liquido di raffreddamento). Complessivamente avremo quindi 7 ingressi e 6 uscite. Iniziamo a configurare il numero ed il nome dei terminali di ingresso e di uscita impostando come segue la prima riga della Matlab Function

```
function [Cxdot,Cpdot,Csdot,Co2dot,Trdot,Tagdot] = fcn(Cx,Cp,Cs,Co2,Tr,Tag,Fag)
```



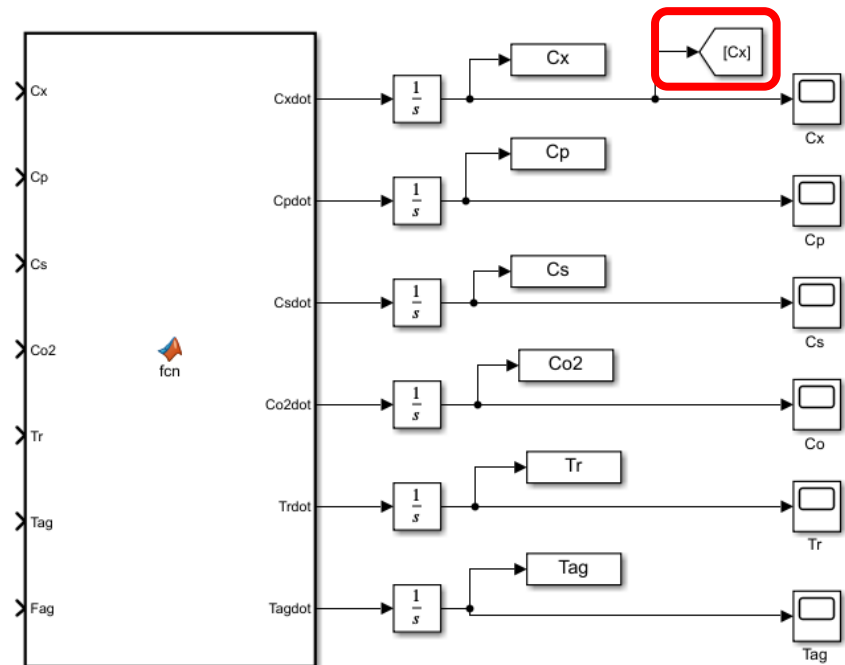
Colleghiamo i terminali di uscita ai terminali di ingresso degli integratori

Ora importiamo una coppia di blocchi «Goto» e «From» per realizzare un collegamento «wireless» fra il segnale

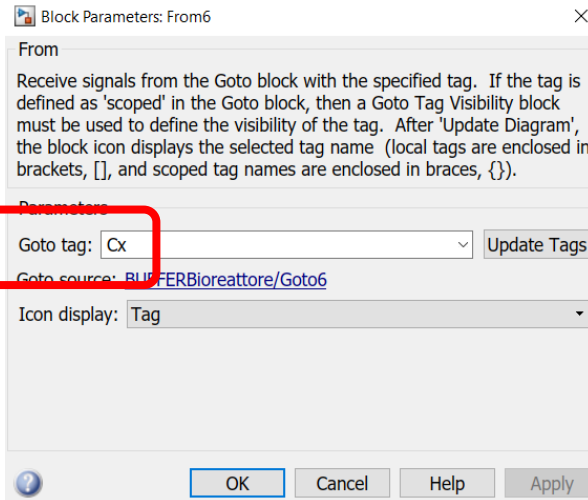


Parametrizzazione del blocco «Goto» mediante l'impostazione di un tag

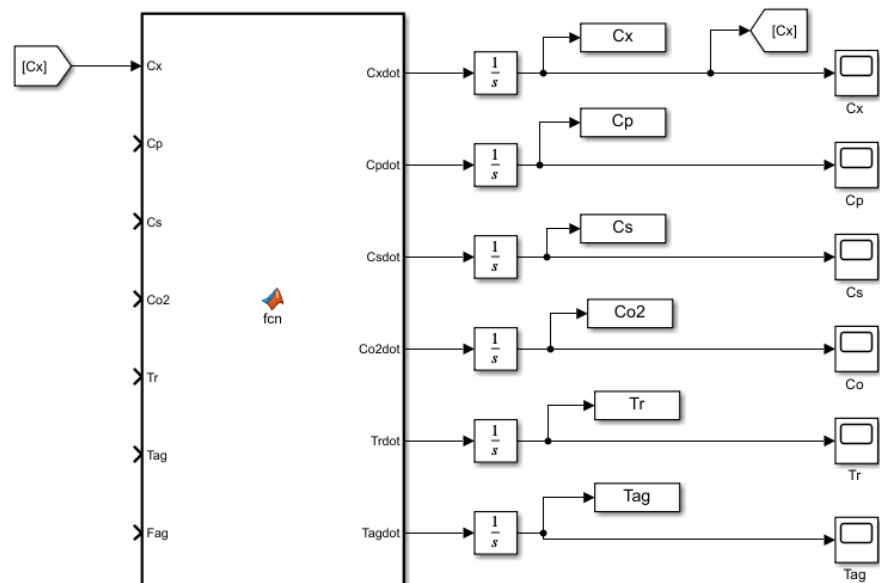
Colleghiamo in ingresso al blocco Goto il segnale al quale desideriamo associare tale tag (ovviamente sarà il segnale $C_X(t)$, disponibile all'uscita del primo integratore in alto)



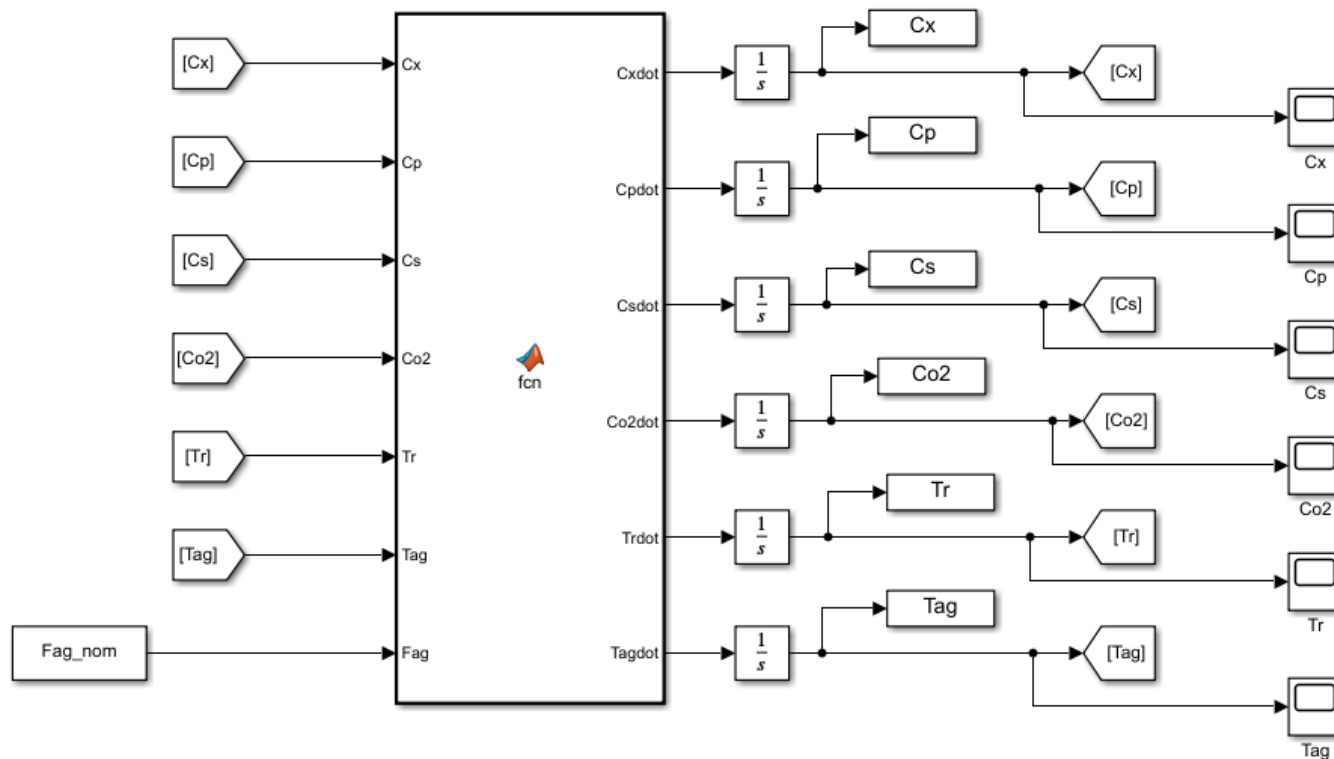
Fatto ciò, in qualunque punto dello schema si desideri accedere al segnale $C_X(t)$ è sufficiente inserire un blocco «From» e configurarlo con il relativo tag, e quindi accedere al segnale mediante il terminale di uscita del blocco From.



Collegiamo l'uscita del blocco From al primo ingresso del blocco Matlab Function



Ora ripetiamo l'operazione per le rimanenti 5 variabili di processo.
Collegiamo anche all'ultimo ingresso del blocco «Matlab Function» un blocco «Constant» che contenga la costante simbolica F_{ag_nom}



Resta da completare il contenuto del blocco Matlab Function.

Poichè all'interno del blocco non sono accessibili le variabili del workspace, queste dovranno essere esplicitamente definite come parte del corpo della funzione. Il corpo della funzione avrà la seguente struttura:

```
function [Cxdot,Cpdot,Csdot,Co2dot,Trdot,Tagdot] = fcn(Cx,Cp,Cs,Co2,Tr,Tag,Fag)
```

```
%DEFINIZIONE DELLE COSTANTI SIMBOLICHE
```

```
%Specific ionic constants [1/g ion]
```

```
HNa=-0.550;
```

```
HCa=-0.303;
```

```
ETC ETC.
```

```
% DEFINIZIONE DELLE VARIABILI DI USCITA
```

```
Cxdot=
```

```
Cpdot=
```

```
Csdot=
```

```
Co2dot=
```

```
Trdot=
```

```
Tagdot=
```

Si completi il corpo della funzione definendo tutte le variabili simboliche e definendo quindi, in funzione di queste e dei segnali acquisiti in ingresso, le espressioni delle sei variabili di uscita del blocco.

Il blocco Matlab Function correttamente configurato è presente all'interno del modello `Bioreattore.slx`

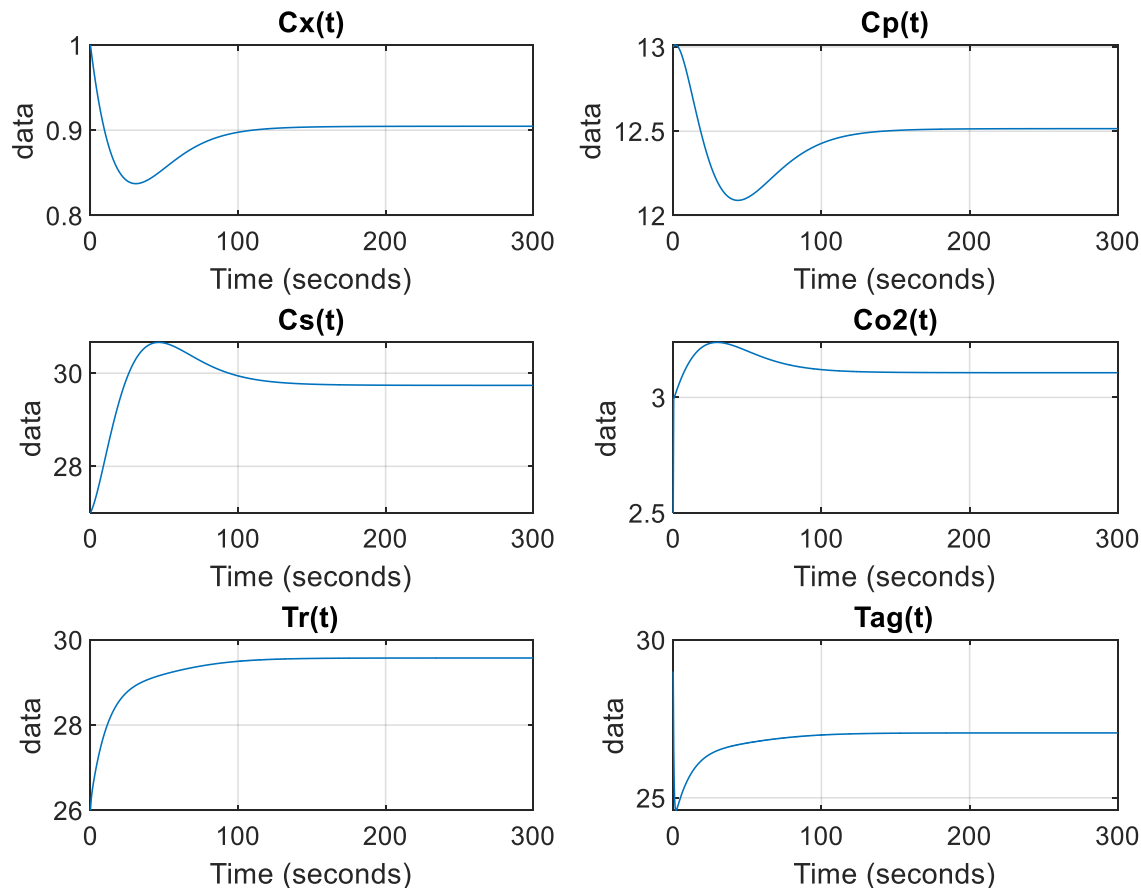
Istruzioni eccessivamente lunghe possono essere suddivise su più righe adiacenti inserendo tre punti alla fine di ogni riga di codice che deve continuare in quella successiva.

Lo script `Bioreattore_dati.m` contiene l'istruzione di run del modello Simulink

```
sim('Bioreattore')
```

ed alcune semplici istruzioni aggiuntive per la creazione di alcuni grafici (le variabili di tipo `TimeSeries` sono passate come unico argomento di ingresso alla funzione `plot`).

Variabili di processo



L'impiego di un solutore a passo variabile fornisce dei grafici analoghi con un peso computazionale inferiore e riducendo del 40% la durata della simulazione rispetto al solutore a passo fisso.

Il seguente codice misura il tempo di esecuzione del modello Simulink e lo scrive sulla command window.

```
tic;  
%tstart = tic;  
sim('Bioreattore')  
%telapsed = toc(tstart);  
toc
```

```
Elapsed time is 3.967165 seconds.  
>> Bioreattore_dati  
Elapsed time is 6.615109 seconds.  
|
```

Test con solutore a passo variabile

Test con solutore a passo fisso (0.01s)

Includiamo nel modello un sistema di controllo automatico che varia in ogni istante la portata $F_{ag}(t)$ del fluido di raffreddamento in modo da portare la temperatura di reazione ad un valore desiderato («set-point») T_r^{des}

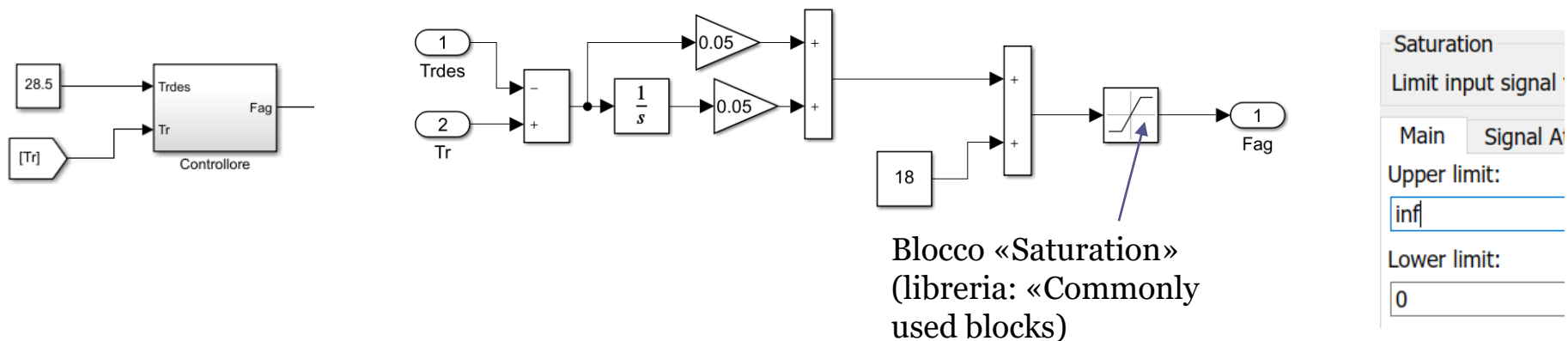
Sistema di controllo automatico della temperatura di reazione

$$F_{ag}(t) = 18 + 0.1[T_r(t) - T_r^{des}] + 0.1 \int_0^t [T_r(\tau) - T_r^{des}] d\tau$$

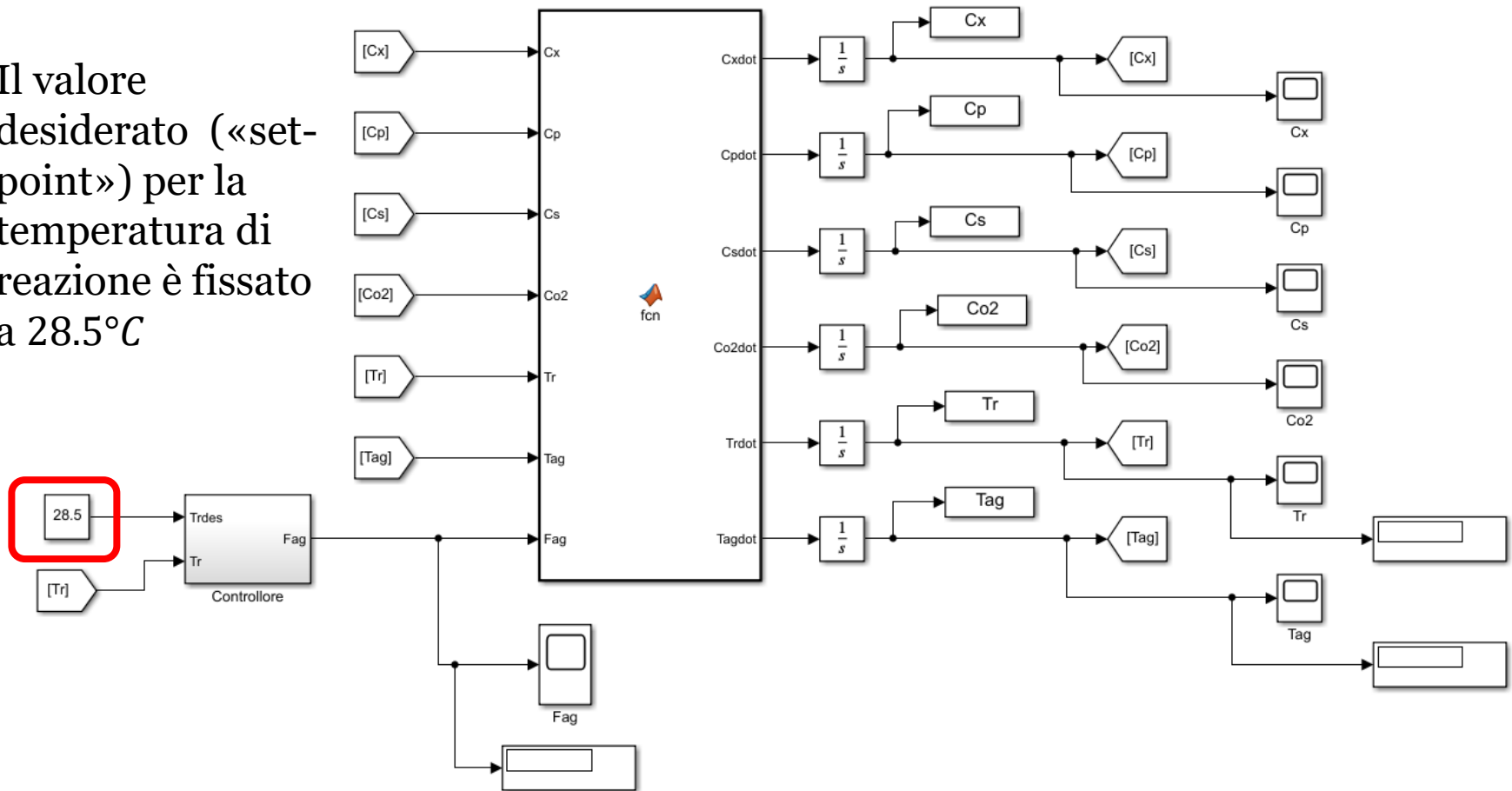
Controllo «Proporzionale-Integrale» con offset costante

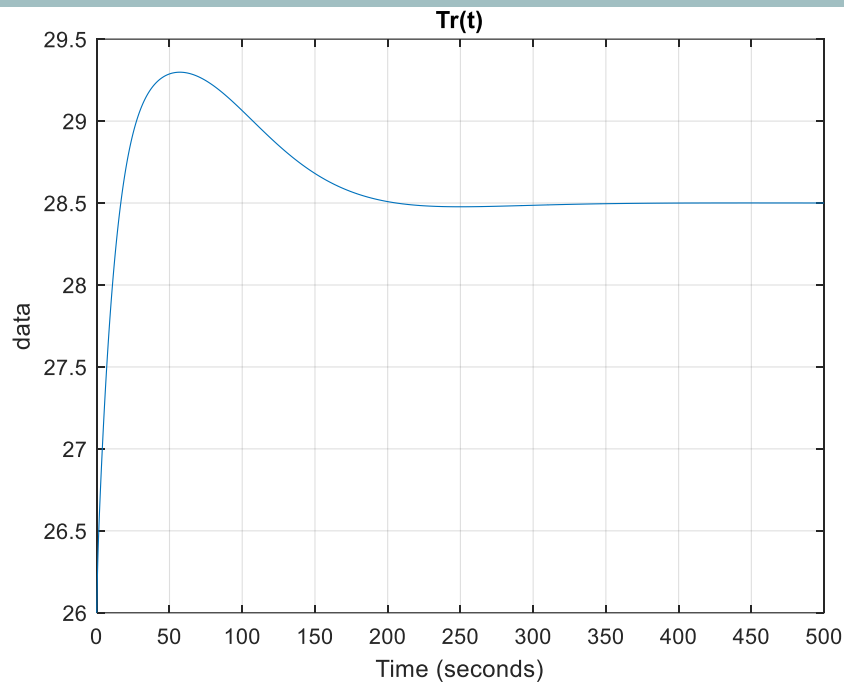
If $F_{ag}(t) < 0$ then $F_{ag}(t) = 0$ La portata non può assumere valori negativi.

Realizziamo un Subsystem «controllore», che riceve in ingresso la temperatura di reazione $T_r(t)$ ed il relativo valore desiderato T_r^{des} e fornisce in uscita la portata $F_{ag}(t)$

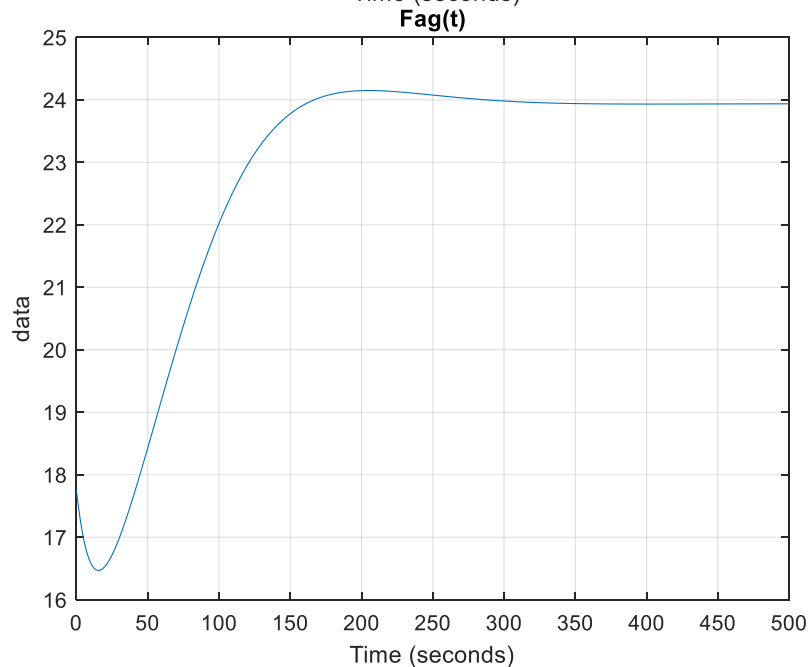


Il valore desiderato («set-point») per la temperatura di reazione è fissato a 28.5°C





La temperatura di reazione raggiunge il valore desiderato



Variazione nel tempo della portata del fluido di raffreddamento