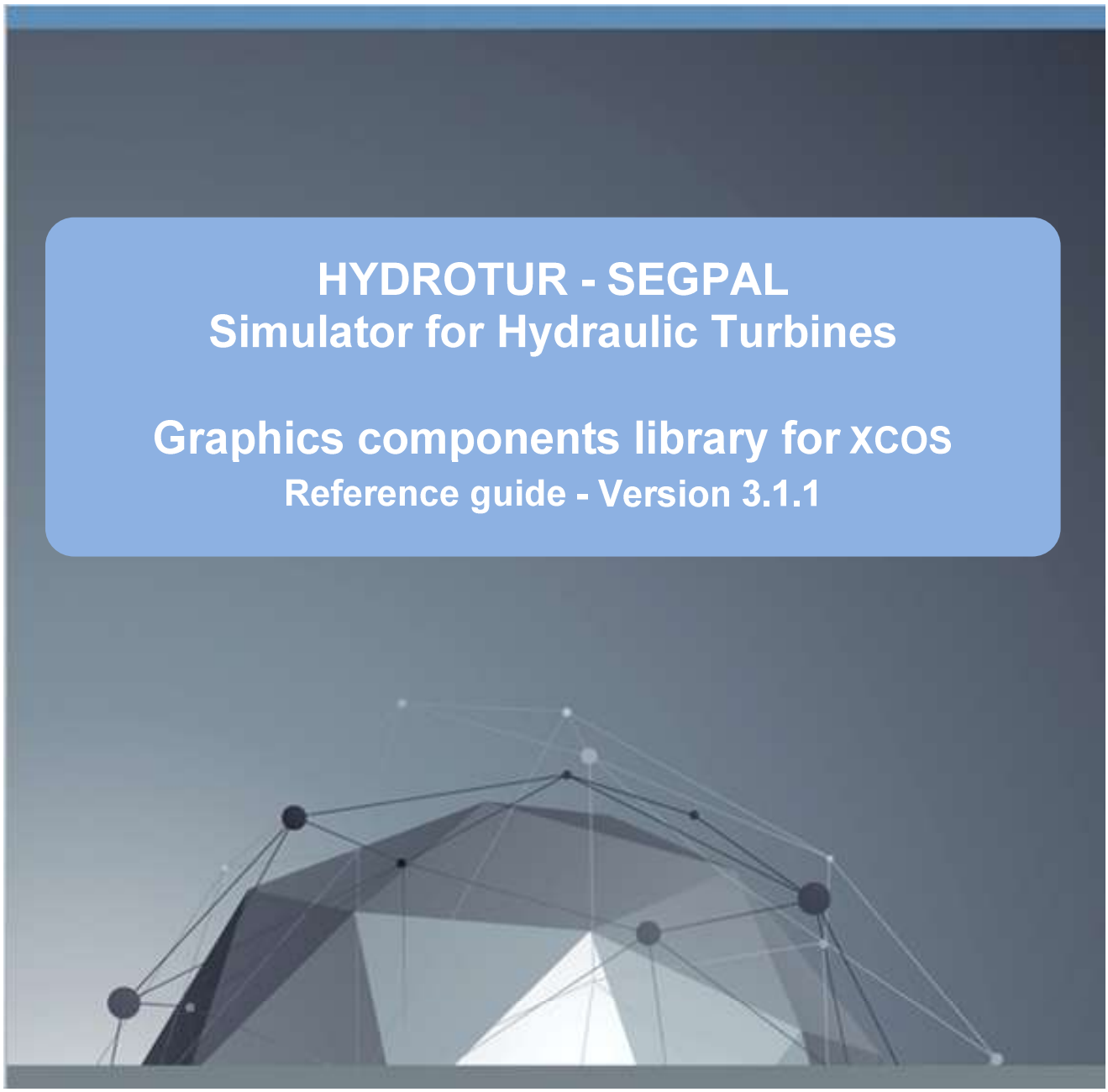
HYDROTUR - SEGPAL
Simulator for Hydraulic Turbines

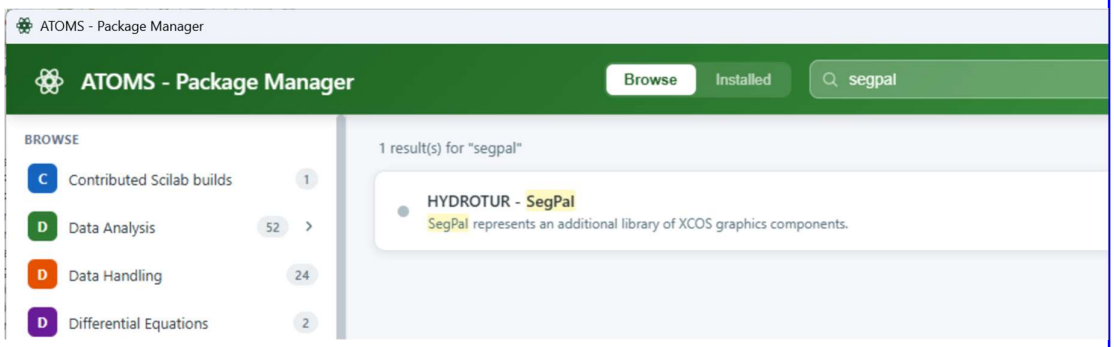
Graphics components library for XCOS
Reference guide - Version 3.1.1



by Pierre Perrichon

Scilab

CHANGES TO THE SEGPAL DOCUMENT

Description	Version
Previous version	2.3
☞ **Scilab versions 2025.1 and 2026.0 have a bug that prevents the correct display of values entered in SegPal function boxes. This bug should be fixed in Scilab version 2026.1. This issue can be tracked at the following address:** **https://gitlab.com/scilab/scilab/-/issues/17505**	
The CLRseg, DerFil, and PSB components also have an Auto mode that allows for automatic initialization based on their input values. Thus, in Auto mode: • The output of the CLRseg and PSB low-pass filters is automatically initialized to the component's input value • The output of the DerFil derivative filter is automatically initialized to 0	3.0.0
SegPal 3.1.1 is compliant with the new module manager—ATOMS—in Scilab 2027 , specifically regarding the management of SegPal's help pages.  However, the pages concerning the Module Manager – ATOMS for Scilab 2027 are not updated in this document. ...The help pages for SegPal components are installed only for Scilab versions >= 2027 and mingw>=10.3.1-3 .else load ATOMS Module SegPal 3.0.1 and mingw 10.3.1	3.1.1

CONTENTS

BIBLIOGRAPHY **4**
TECHNICAL SUPPORT - CONTACTUPPORT TECHNIQUE - CONTACT **4**
LIST OF HYDROTUR DOCUMENTS **5**
FOREWORD **6**
INDEX **7**
1 INTRODUCTION **8**
2 SUPPLY **9**
2.1 Installing SegPal 9
2.2 Uninstalling SegPal 10
3 CONSTRUCTION AND LOADING OF THE SEGPAL LIBRARY **11**
3.1 Foreword 11
3.2 Prerequisites – Installation of the MinGW compilation toolchain in Scilab – ATOMS modules 11
3.3 The builder.sce constructor 13
3.4 The loader « loader.sce » 13
3.5 Direct access to the SegPal installation directory 13
4 ARCHITECTURE OF TEST PROGRAMS **14**
4.1 Selection and configuration of the Runge-Kutta XCOS solver 14
4.1.1 Solver Selection 14
4.1.2 Simulation time settings 14
4.2 Graphic outputs 15
4.2.1 Principle of numbering graphics windows 15
4.2.2 Managing legends associated with a graphics window 15
4.3 Launching a test program 17
5 SPECIFIC COMPONENTS OF THE SEGPAL LIBRARY **18**
5.1 Panoramic view of the additional HYDROTUR– SegPal components 18
5.2 Description of components 18
6 ONLINE HELP **55**
6.1 Get online help for each component 55
6.2 Language change 55
APPENDICES **56**
A. LISTING TYPE AFTER BUILDER'S EXECUTION **56**
LIST OF FIGURES **57**

BIBLIOGRAPHY

[1] Scilab Enterprises - Philippe Roux, Scilab - De la théorie à la pratique - I.Les fondamentaux, D-Booker, 2013.

[2] Scilab Enterprises - Yvon Degré, Serge Steer, Scilab - De la théorie à la pratique - II-Modéliser et simuler avec XCOS, D-Booker, 2014.

[3] Scilab Enterprises - Laurent Berger, Scilab - De la théorie à la pratique - III. Le traitement du signal, D-Booker, 2014.

[4] IEC 60308, Turbines hydrauliques – Essais des systèmes de régulation, 2005.

[5] IEC 61362, Guide pour la spécification des systèmes de régulation des turbines, 2012.

[6] L. VIVIER, Turbines hydrauliques et leur régulation, PARIS: Editions Albin Michel, 1966.

[7] Y.Granjon, AUTOMATIQUE - Systèmes linéaires, non linéaires, à temps continu, à temps discret, représentation d'état, NANCY: DUNOD - SCIENCES SUP, 2010.

[8] E.GODOY, Régulation industrielle, L'usine Nouvelle Série EEA, 2007.

[9] P. De LARMINAT, Automatique des systèmes linéaires - Tome 3 - Commande, Flammation Sciences, 1977.

[10] P. PERRICHON, “Eléments de synthèse pour la réalisation d'un régulateur de vitesse numérique industriel dans la conduite d'un groupe hydraulique,” Grenoble Université - INP - ENSE3, Grenoble, 2011.

[11] B. GREIVELDINGER, “Etude de la stabilité de réglage d'une centrale hydroélectrique en réseau séparé,” Ecole polytechnique de LAUSANNE, LAUSANNE, 2003.

Document type	References
Scilab Engineering	[1] [2] [3]
IEC normative references	[4] [5]
Scientific literature	[6] [7] [8] [9]
University documents	[10] [11]

TECHNICAL SUPPORT - CONTACTUPPORT TECHNIQUE - CONTACT

Mr Pierre PERRICHON
Ingénieur diplômé par l'Etat
Automatique & Informatique industrielle

Projet HYDROTUR

perrichon.pierre@wanadoo.fr

Figure 1 : Technical support for the HYDROTUR - SEGPAL project

LIST OF HYDROTUR DOCUMENTS

Document Title
HYDROTUR – HydroSim - Installation and User Guide
HYDROTUR – HydroSim - Programming Guide
HYDROTUR – HydroSim - Schematic libraries of FRANCIS-KAPLAN-PELTON models and circuits
HYDROTUR – HydroStab Solution – Stability Studies – User Guide
HYDROTUR – HydroSim CALTUR- Calculation note CALTUR - Version 2.4 (en-fr) .xlsx
HYDROTUR – SEGPAL– Graphics components library for XCOS - Reference Guide

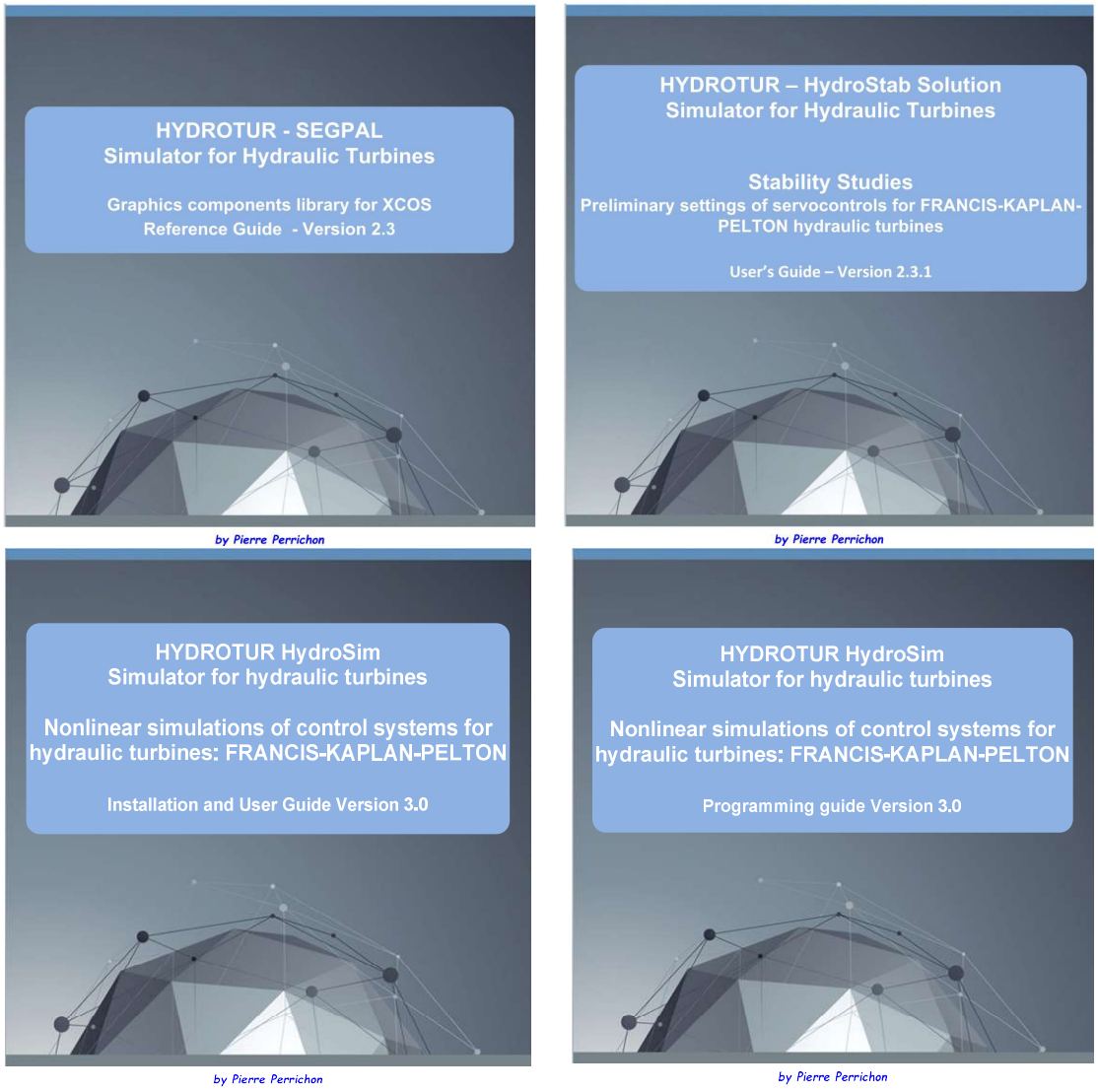


Figure 2 : List of main HYDROTUR documents

FOREWORD

The **HYDROTUR** product essentially brings together three software components created to help simulate rotating machines in the field of hydroelectricity and is based on the study of hydraulic turbines of the FRANCIS, KAPLAN, PELTON type.
Thus, **HYDROTUR** includes the following products:

- **SEGPAL**, an extension library for the graphical components of Scilab-Xcos
- **HydroStab,** dedicated to calculating the stability of a speed regulator controlling a hydroelectric machine, and carried out with linear simulations in small motion of the processes to be analysed.
- **HydroSim,** an extension of the **HydroStab** product, is characterized by non-linear simulations of the processes studied, in all operating modes of the turbines.

These components are written in the Scilab-Xcos environment and are all available and downloadable as ATOMS library modules for Scilab.

INDEX

C

CLRseg 19

D

DerFil 20
DiracSync 22

E

EdgeTrigger 24

G

GainVar 27
GeneBF 30

H

Hysteresis_H 32

I

Interpln_H 33
IntLim 34
IntSPE 36

M

Memo 37

P

Pow 38
PSB 40

R

Rampe 41
Relais_H 43
RELATIONOP 46
Répartiteur_Jets1.zcos 45
Répartiteur_Jets2.zcos 46

S

Satur 47
SinuSync 48
Step2 50

T

TempoTrig 51
Time Delay 53

1 INTRODUCTION

SEGPAL represents an additional library of XCOS graphics components.

SEGPAL harmoniously complements the standard XCOS tool palettes, and has been specially designed to improve the ergonomics and relevance of **HYDROTUR-HydroSim** schematic libraries and simulators.

SEGPAL is initially dedicated to the **HydroSim** project, simulators for FRANCIS, PELTON, KAPLAN hydraulic turbines entirely produced under XCOS but can be used in any other XCOS program as soon as the SEGPAL library has been loaded.

Some components are designed to enable steady-state simulations of processes and therefore avoid undesirable transient conditions at the start of the simulation.

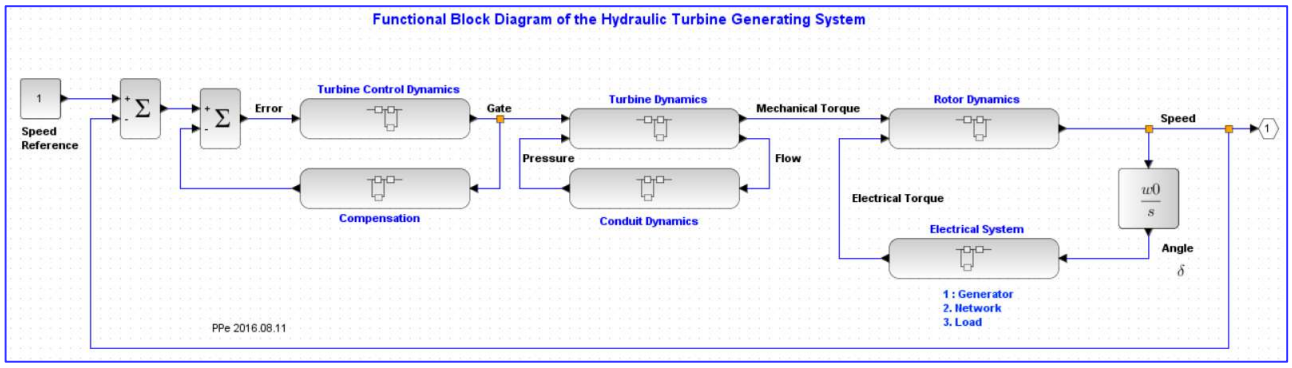


Figure 3 : General functional diagram of a hydroelectric control system

☞ **If SegPal is not loaded, it is not possible to use the simulated turbine models, or to run the demonstration programs integrated in this delivery.**

This document describes all the components present in the SEGPAL library.

Each component is associated with a basic XCOS test program, giving an example of its use.

The table below summarizes the main components developed within this library:

Component	Meaning	SegPal\demos
CLRseq	First order low pass filter with initialization at t=0	CLRseg.zcos CLRseg_Auto.zcos
DerFil	Filtered derivative with initial condition	DerFil.zcos DerFil_Auto.zcos
DiracSync	Generation of a Dirac's comb synchronized to a start date	DiracSync.zcos
EdgeTrigger	Edge detector of an input square wave	EdgeTrigger.zcos
GainVar	Variable gain - Adaptation to the simulation of multi-jet PELTON machines	GainVar.zcos GainVar_V.zcos
GeneBF	Low frequency signal generator - Signal disruptor	GeneBF.zcos
Hysteresis_H	Multivariable hysteresis	Hysteresis_H.zcos
Interpln_H	Multivariable linear interpolation	Interpln_H.zcos
IntLim	Pure integrator with reset and external saturations	IntLim.zcos
IntSPE	Special integrator with initial conditions and quick desaturation	IntSPE.zcos
Memo	Pure delay	Memo.zcos

Pow	Raising a positive number to the power	Pow.zcos
PSB	1st order low-pass filter with reset and external saturations	PSB.zcos
Rampe	Integrator ramp with reset and external saturations	Rampe.zcos
Relais_H	Multiple relays in parallel on a 1x1 input signal	Relais_H.zcos Repartiteur_jets1.zcos Repartiteur_jets2.zcos
RELATIONALOP_H	Multi-form relational operations	RELATIONALOP multivariables.zcos
Satur	Multivariable external saturations	Satur.zcos Satur n lignes.zcos
SinuSync	Sine function with synchronization at the start time	SinuSync.zcos Sinus amorti.zcos
Step2	Double programmable duration step	Step2.zcos
TempoTrig	Retriggerable timer	TempoTrig.zcos
Time_Delay	Input time offset	Time_Delay.zcos

Figure 4 : SegPal Library Components

2 SUPPLY

2.1 Installing SegPal

The **HYDROTUR-SegPal** project is provided from a “SegPal.zip” file, which must be unzipped.

The package can be installed in different ways:

Mode	Method
1.	Using the Scilab ATOMS module installer, pointing to HYDROTUR-SegPal in the "Xcos-Palettes" module
2.	By entering the command atomsInstall("SegPal") in the Scilab console
3.	Manually, by placing and extracting the ZIP file into a user-defined directory, such as "HYDROTUR\SegPal\" for example (accessible at https://atoms.scilab.org/toolboxes/SegPal/3.0.0)

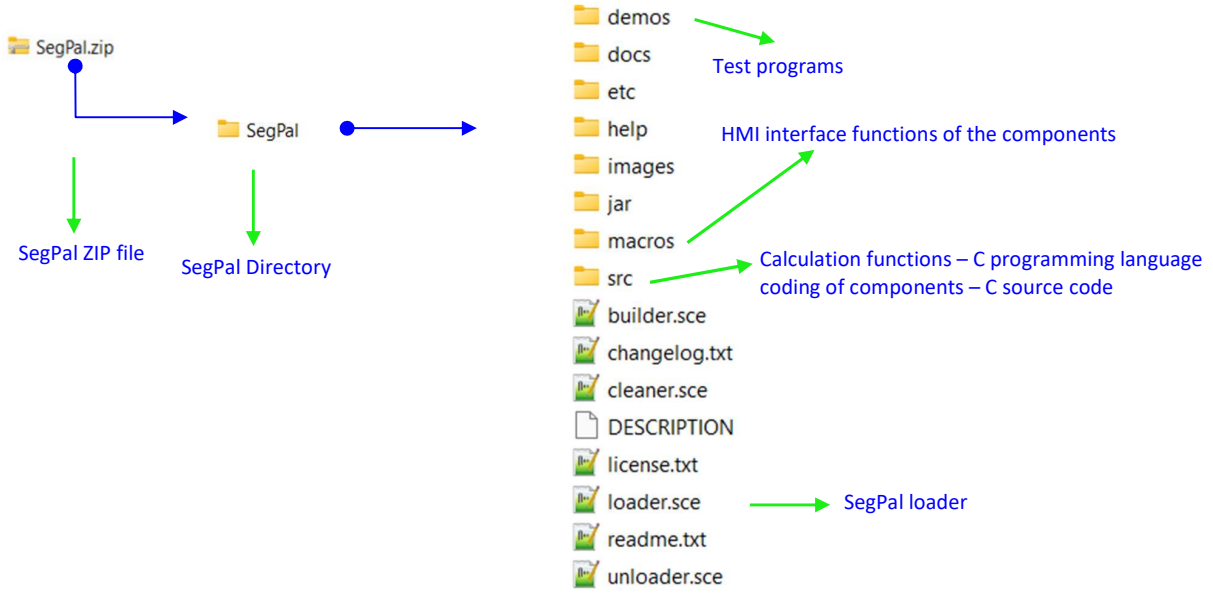


Figure 5 : SEGPAL directory organization

Directory or File	Content
demos	Set of demonstration programs illustrating each SegPal component
Docs	Documentation and installation guides (English/French):
Etc	Initialization (SegPal.start file) and finalization (SegPal.quit file). Scripts used to load the SegPal module (loader.sce) or unload it (unloader.sce).
Help	Bilingual online help for each component Online help can be accessed by right-clicking on a component in the library (Français/English)
macros	Macros (Files with the ".sci" extension) Set of source files written in Scilab, describing the GUI interface of the components
src\c	The complete set of source files written in C language. Execution code for each component.
builder.sce	Main builder of the SegPal module.
changelog.txt	SegPal version change history
cleaner.sce	Delete the compiled ".bin" and ".o" files in the "macros" and "src\c" directories. Use this with any rebuild using the "builder.sce" utility.
DESCRIPTION	Brief description of SegPal. Used when loading SegPal locally. For Designer use only.
licence.txt	Declaration of conformity to the CeCILL license Version 2.1
loader.sce	Loading SegPal into Scilab. This file is created automatically when the SegPal library is regenerated using the "builder.sce" constructor.
readme.txt	Recommendation for installing SegPal on the computer
unloader.sce	Unloading HydroSim into the Scilab platform. This file is automatically created when the HydroSim library is regenerated using the "builder.sce" constructor.

Figure 6 : Main files in the SegPal directory

2.2 Uninstalling SegPal

In mode 1 or 2 (*see Figure 5)*, SegPal can be uninstalled by clicking on the Scilab ATOMS icon. , or by typing the command **atomsGui()** in the Scilab console, then selecting the SegPal module:

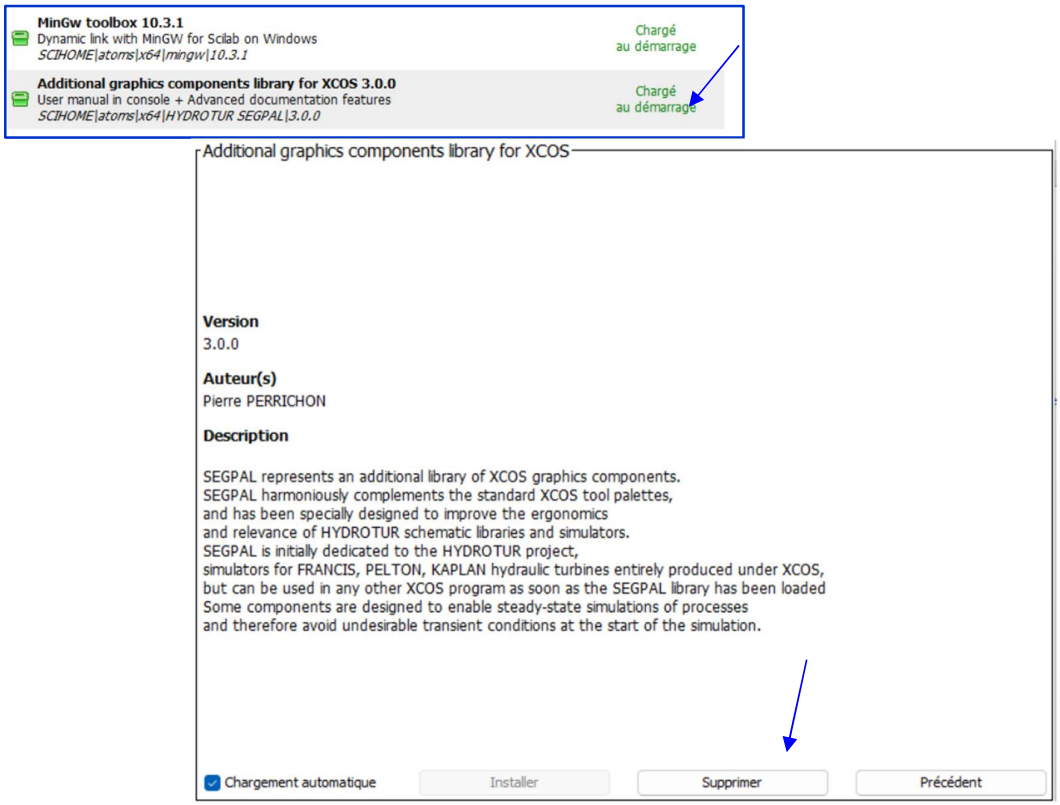


Figure 7 : Uninstalling SegPal

3 CONSTRUCTION AND LOADING OF THE SEGPAL LIBRARY

3.1 Foreword

The construction of the **SegPal** library requires a good understanding of the book [2][1].

Its use requires only knowledge of this manual "HYDROTUR - SEGPAL - Reference Guide", and the components it incorporates.

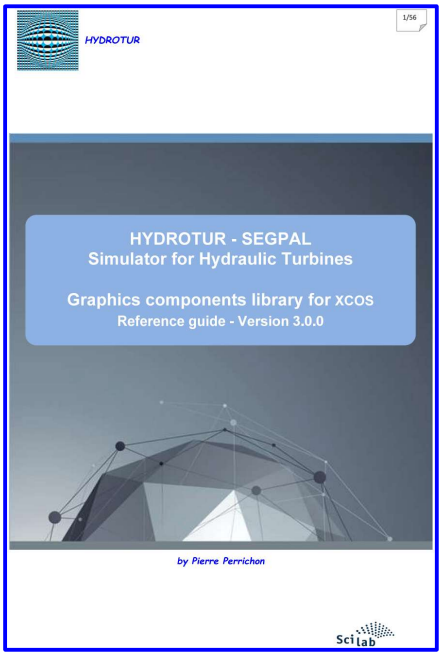




Figure 8 : « Modéliser et simuler avec XCOS » book

3.2 Prerequisites – Installation of the MinGW compilation toolchain in Scilab – ATOMS modules

The source code for the executables related to the functional boxes is mainly written in C language and is available in the directory SegPal\scr\c.

Therefore, it is necessary to have a GNU C/C++ compiler installed on the user's machine.
To do this, launch the ATOMS module manager in Scilab and select the MinGW module from the list of modules:

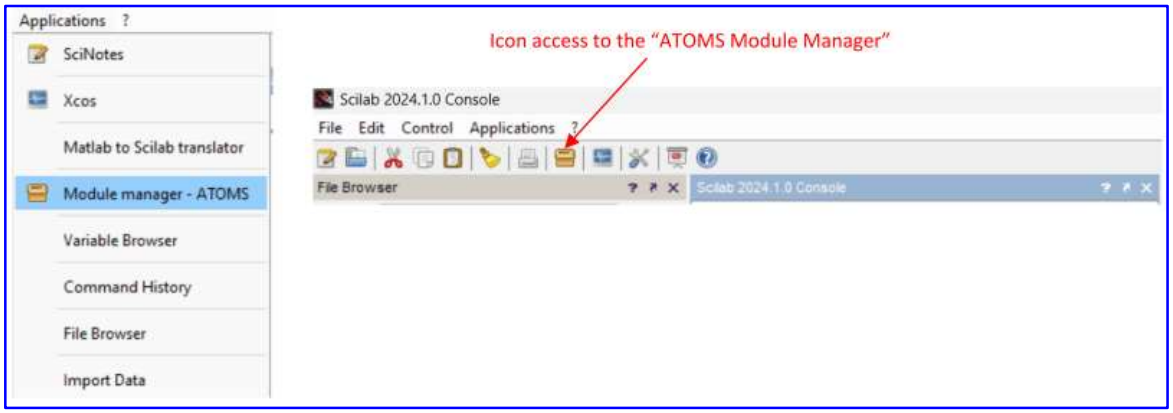


Figure 9 : Access to ATOMS Modules in Scilab libraries

[1] This book can be ordered online from D-Booker publishers.

1. Click on the "All modules" icon to access the "Main categories" view.



2. In the drop-down list sorted alphabetically, select the "MinGW" component:

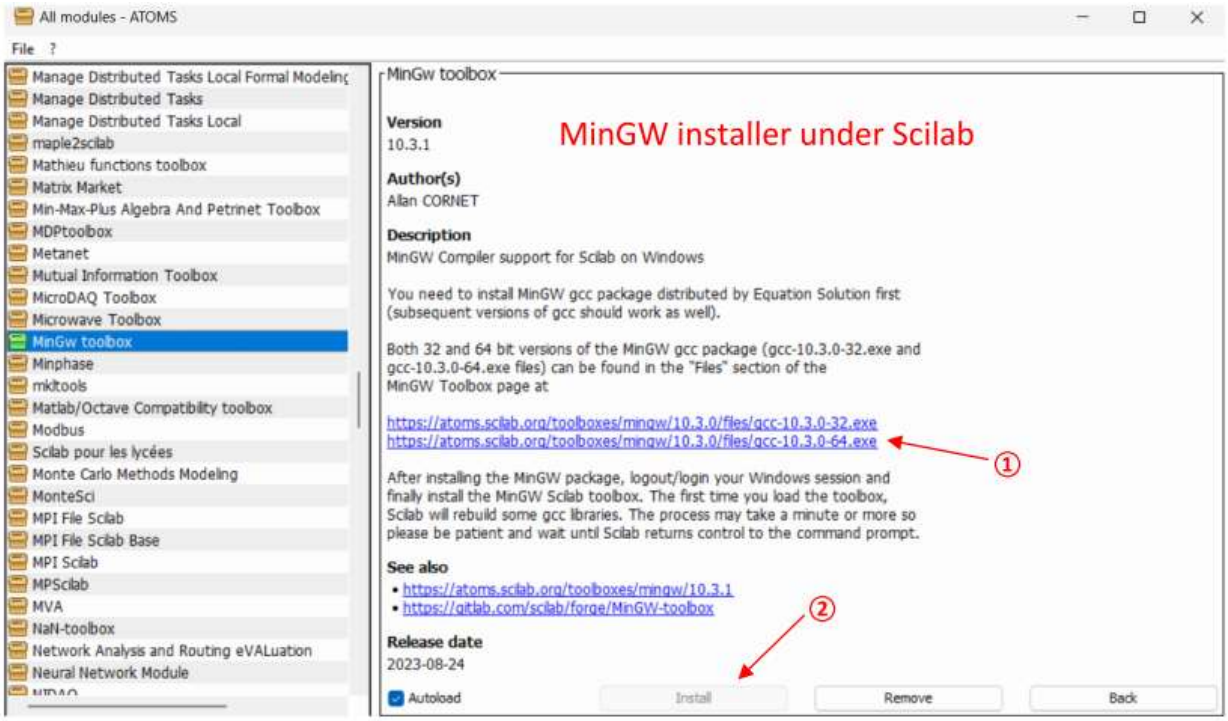


Figure 10 : MinGW installer under Scilab

3. Retrieve the 64-bit MinGW compiler in the version indicated in ① and install it in the directory chosen by the operator:

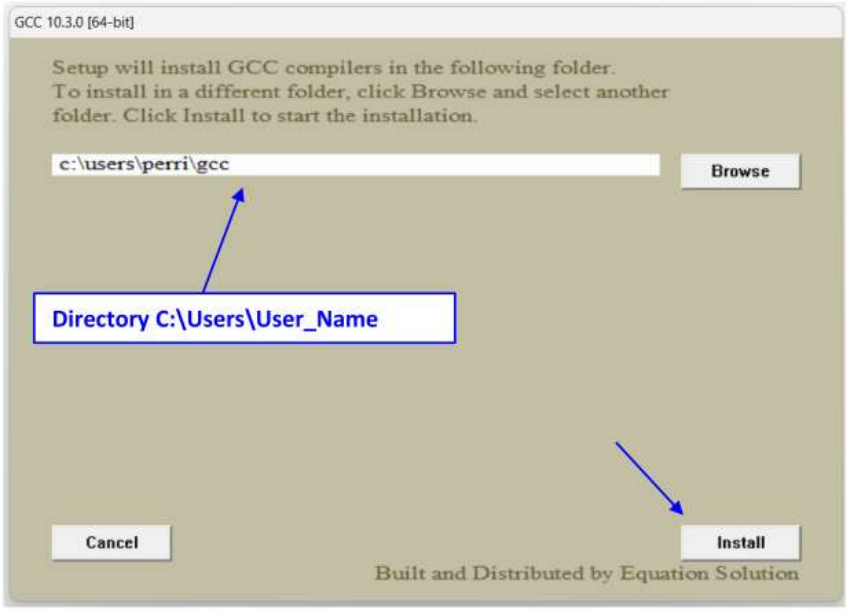


Figure 11 : Installing the gcc compiler

Generally, unless the operator is modified, a default directory is created on "C:\Users\User_Name".

At the end of the installation, you can check that "gcc" is correctly installed:

- Open a new Scilab session
- Enter the command **powershell("gcc --version")** in the Scilab console.

```
--> powershell("gcc --version")
 ans  =

  "gcc.exe (GCC) 10.3.0"
```

Figure 12 : Checking the installation of the GCC compiler-linker, version 10.3.0 64-bit

4. Finish the MinGW installation by clicking on "Install", as indicated in ② on *Figure 10*
5. Close the Scilab session and start a new session to complete the installation.

3.3 The builder.sce constructor

The script builder.sce file, located in the SegPal directory, creates the SegPal library and provides all the files necessary for its use:

- Compiling all .sci macros associated with the macros subdirectory of SegPal\macros
- Compiling all the C source code associated with the function boxes, in the SegPal\src\c directory
- Creation of help files in French and English, allowing online help on each component by simply right clicking on a component.

In addition, the builder.sce constructor provides and adds the following files to the SegPal directories:

- cleaner.sce Cancels and deletes all files created by the builder.sce constructor
- loader.sce Loading the SegPal library in Scilab.
- unloader.sce Unloading the SegPal library into Scilab

A listing, obtained by screenshot after launching the builder.sce in Scilab, is provided in *Appendix A*.

3.4 The loader « loader.sce »

The **loader.sce** is created by the **builder.sce** interface, as shown in §3.3.

Running this script loads the **SegPal** library into **Scilab**.

If successful, the Scilab console displays a list of the operations performed:

```
Start SegPal Version 3.0.0 : Additional components for XCOS graphic library
        Load macros
        Load palette
        Load simulations functions
        Load help

-->
```

Figure 13 : Loading the SegPal library into Scilab – Using the loader.sce file

3.5 Direct access to the SegPal installation directory

To directly access the Segpal installation directory, shown in *Figure 5*, enter the following two instructions in the Scilab console:

```
global SCI_SegPal
chdir(SCI_SegPal)
```

4 ARCHITECTURE OF TEST PROGRAMS

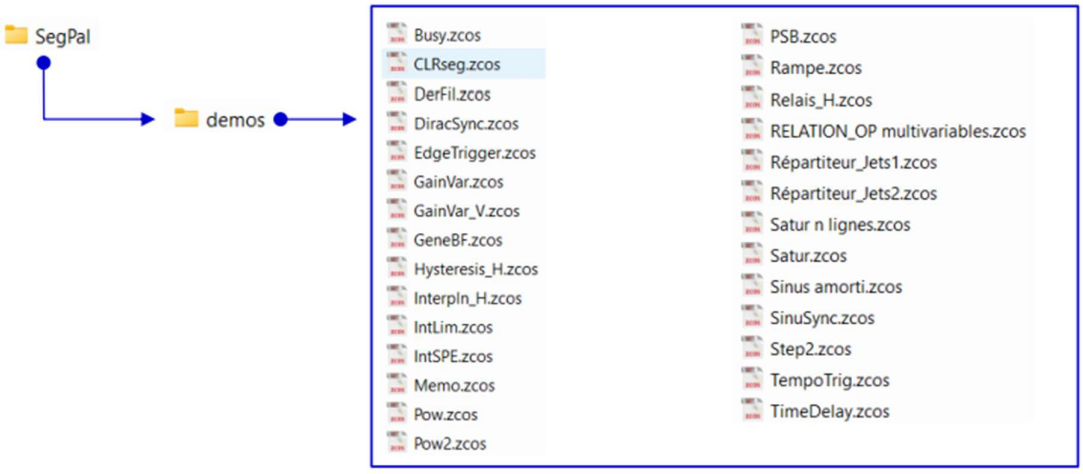


Figure 14 : Overview of the SegPal library's unit testing programs

4.1 Selection and configuration of the Runge-Kutta XCOS solver

4.1.1 Solver Selection

The programs use the default configuration defined by **HYDROTUR HydroSim.**

This device is described in the **HYDROTUR installation and operating manual**, and can be checked by accessing the **Simulation➔Setup** menu of XCOS

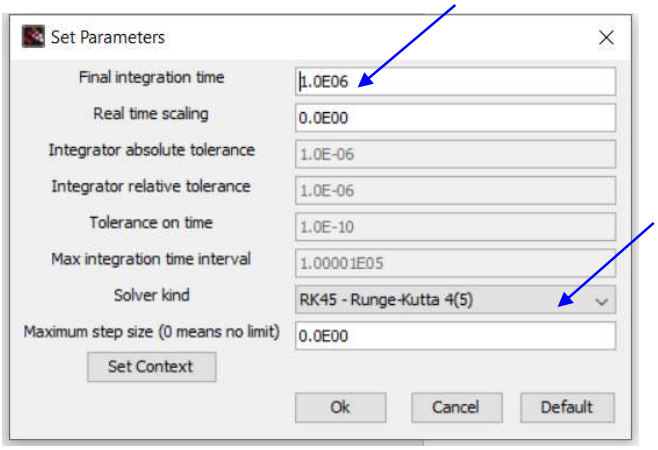


Figure 15 : Runge-Kutta solver settings

The configuration here sets a maximum simulation time of 10^6 s.

4.1.2 Simulation time settings

The actual simulation duration is adjusted in each program using the TSim variable (via **Simulation➔ Set Context** menu).
This variable is used in each **END** block associated with each test program.

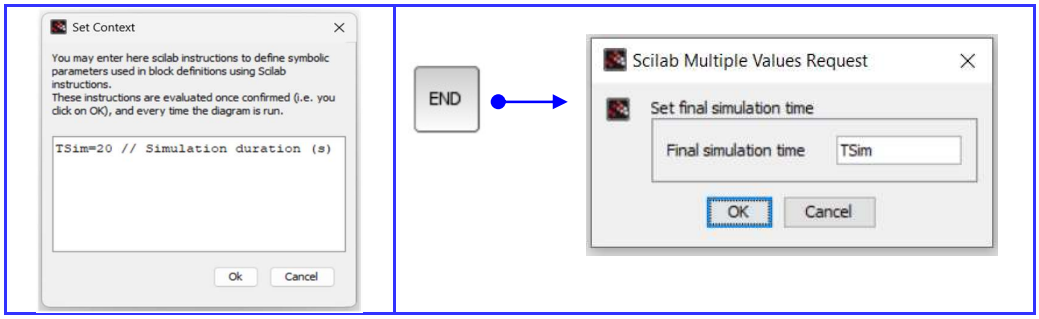


Figure 16 : TSim simulation time setting

4.2 Graphic outputs

4.2.1 Principle of numbering graphics windows

Each test program includes a multiplexed graphical output:

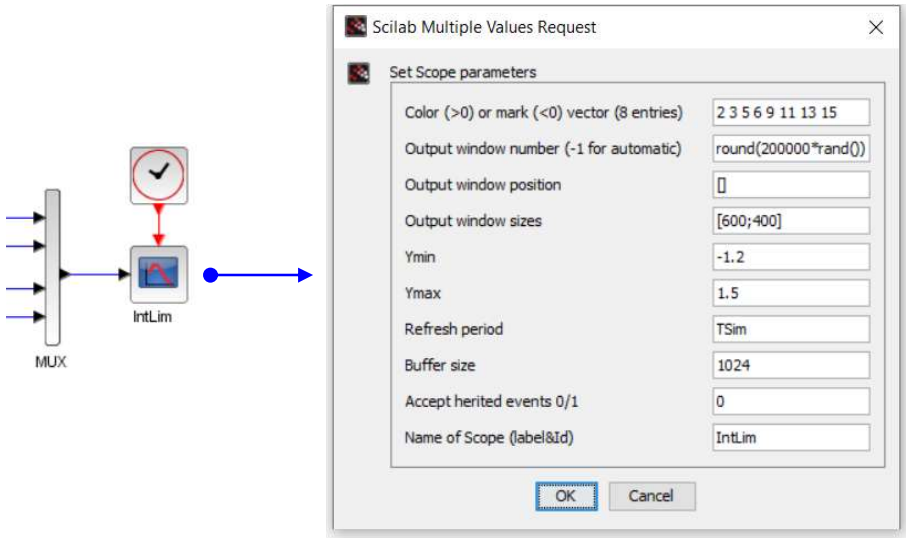


Figure 17 : Principle of numbering graphics windows

To ensure the uniqueness of each graphics window, when launching a test program, the window numbering is generated from the calculation of a random number, based on the formula "**round(200000*rand())**", i.e. the integer rounded to the nearest value of the floating-point number 200000*rand().

4.2.2 Managing legends associated with a graphics window

A legend is automatically inserted into the graphics window of a test program at the end of each simulation run.

This device is generated at time t=TSim, by a sequence of graphical instructions in XCOS, adapted to the test program:

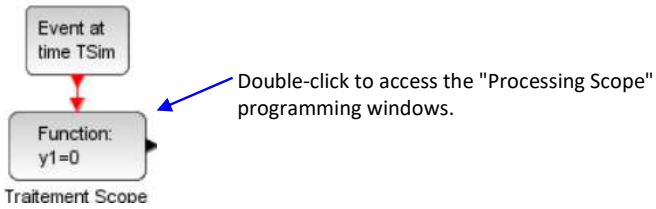


Figure 18 : Principle of legend insertion in test programs

The programming of the “Scope Processing” function is accessible by double-clicking in the “Function” component, and is summarized in the table below:

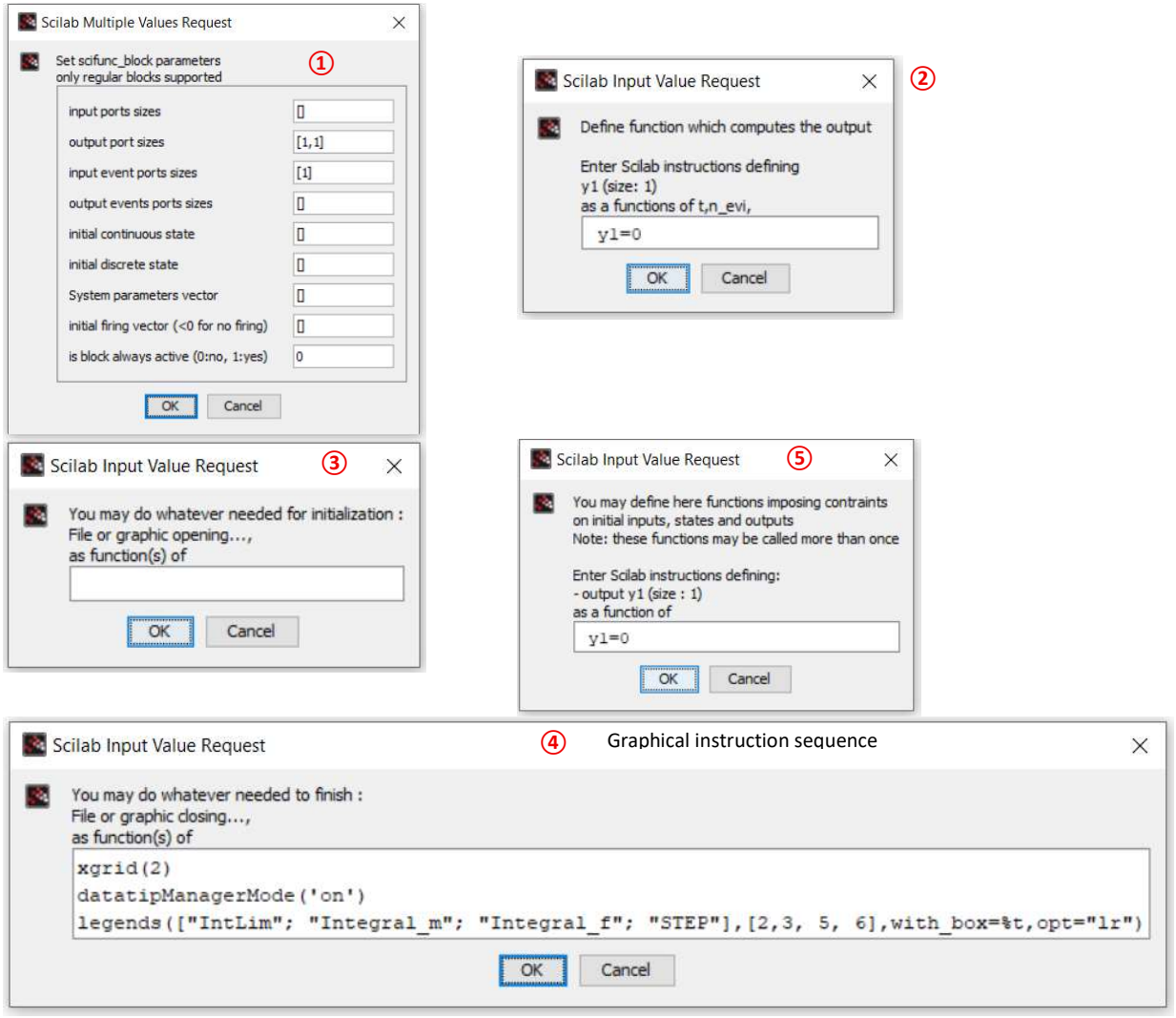


Figure 19 : Programming windows for the "Scope Processing" function– **IntLim** Component

Window ④ provides the instructions necessary to generate a legend in the graph plot window:

xgrid(2)	Add a grid to the blue graphics window
datatipManagerMode(“on”)	Automatic startup of datatips mode in the graphics window
legends(…)	Displayed channels names; Display colours with_box=%t addition of a box surrounding the legend opt=”lr” – Legend position. Here, the legend is located at the bottom (l) right (r) of the graph.

☞ The reader will obtain further information on the “legends” function by entering ‘**help legends’** in the Scilab console, or “**legends Scilab**” in a web search engine.

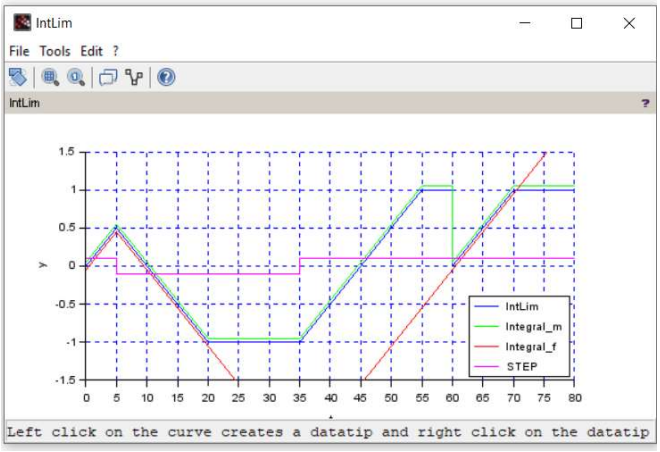


Figure 20 : Graphical window of the test program associated with the **IntLim** component

4.3 Launching a test program

A test program can be run as soon as it has been loaded into an XCOS window.

To do this, position the Scilab file manager in the utilities directory, double-click on the test program to display its contents in an XCOS window:

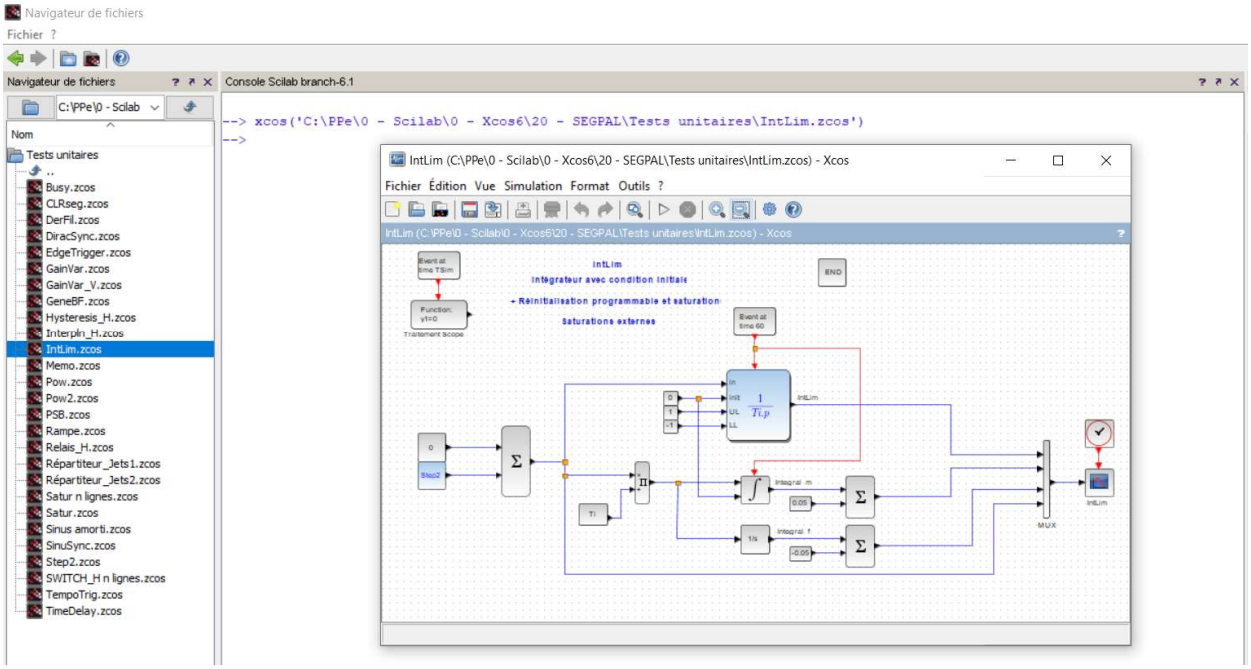


Figure 21 : Loading and running a test program

Unlike launching **HydroSim's turbine simulators**, which generally runs from a "sce" file, the test program runs directly by clicking the dedicated tool in the XCOS menu bar:

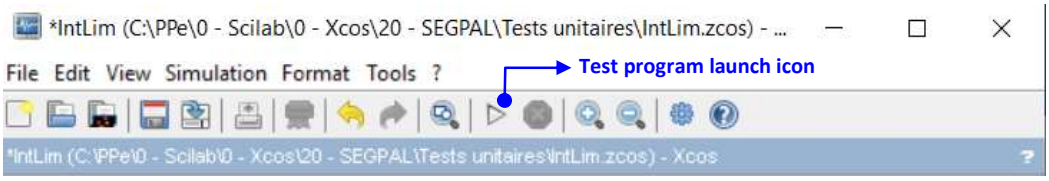


Figure 22 : Test program launch icon

5 SPECIFIC COMPONENTS OF THE SEGPAL LIBRARY

5.1 Panoramic view of the additional HYDROTUR– SegPal components

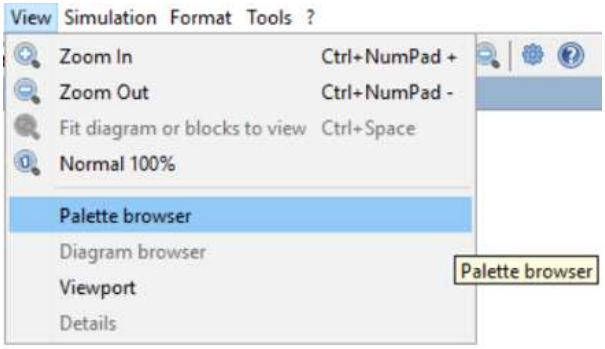


Figure 23 : Accessing the SegPal palette in an XCOS graphics window

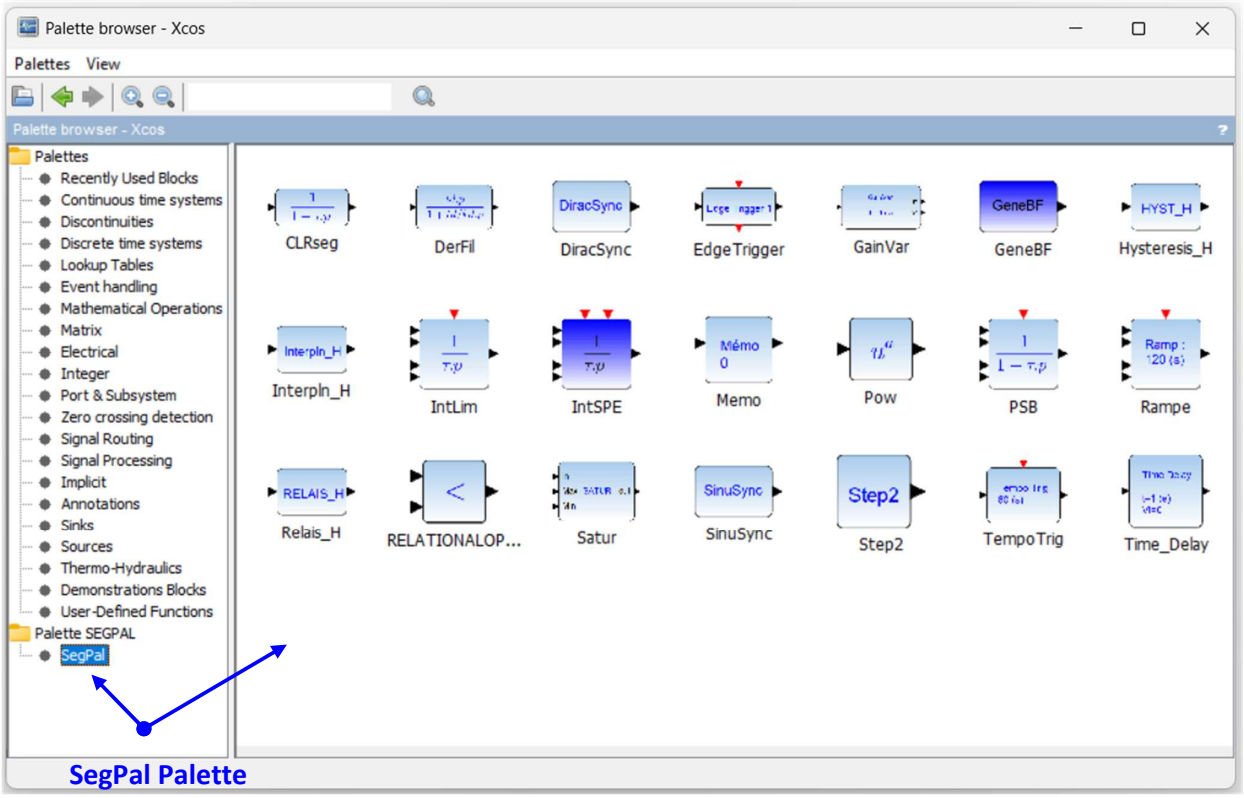


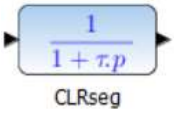
Figure 24 : Panoramic view of the additional **HYDROTUR– SegPal** components

5.2 Description of components

CLRseg

First order low-pass filter with filter initialization at t=0

Block Screenshot

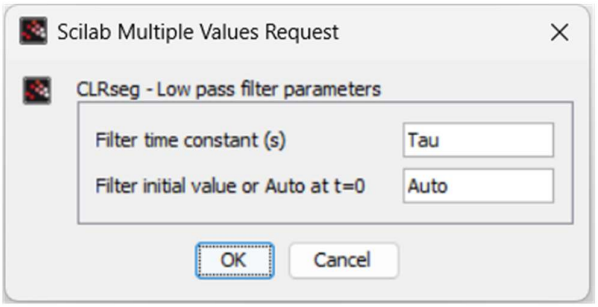


Description

The CLRseg block represents a first-order low-pass filter.
The initial value of the block is programmable

- In normal mode, the CLRseg filter is initialized based on the initial value indicated in its function box.
- In automatic "Auto" mode, the filter is automatically initialized according to its input value and maintains its output equal to its input value.

Parameters



Alternative forms

Interfacing function

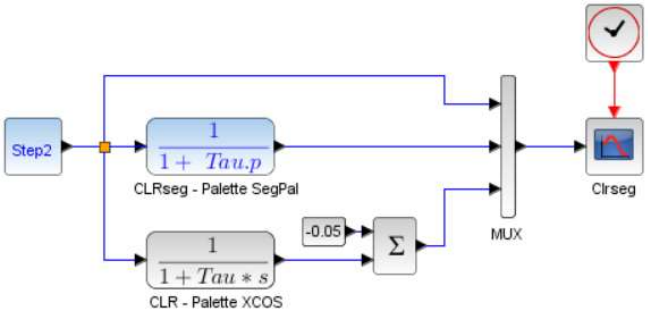
SegPal\macros\CLRseg.sci

Computational function

SegPal\src\c\CLRsegC.c

Test program

- SegPal\demos\CLRseg.zcos



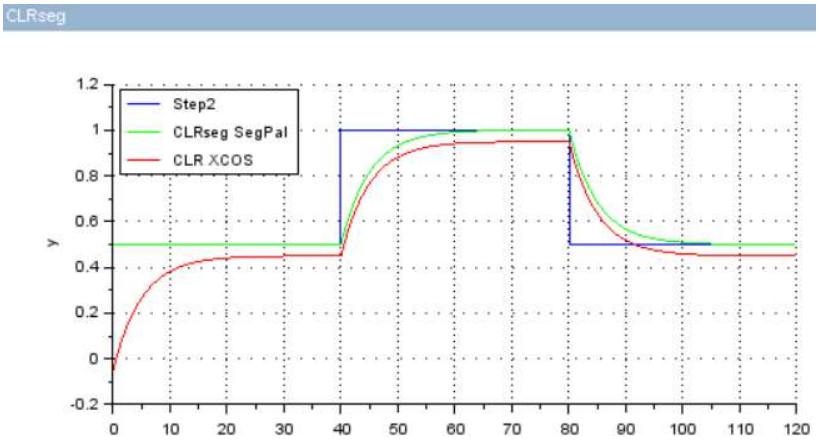
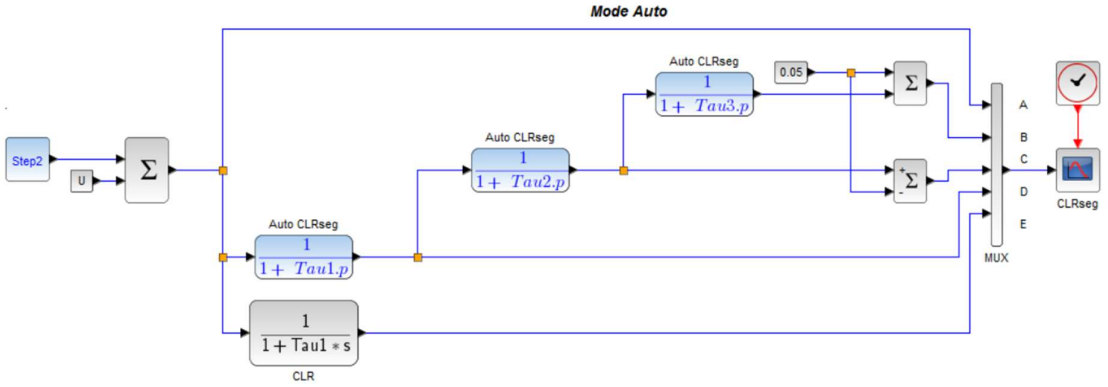


Figure 25 : CLRseg.zcos test program

- SegPal\demos\CLRseg_Auto.zcos



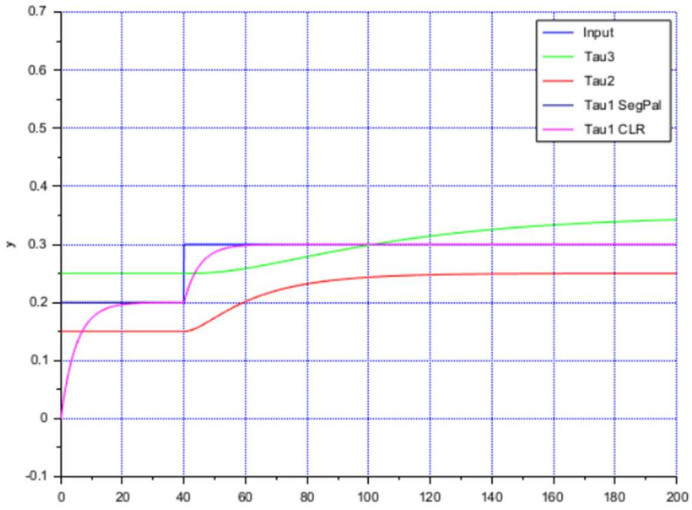
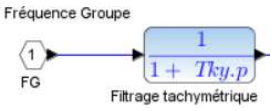


Figure 26 : CLRseg.zcos_Auto test program

Notes



The CLRseg block is, for example (see **HydroSim**), used in REGxx models of the cruise control system to filter the speed of the group.
Depending on the state of the circuit breaker, the initial value is 0 if the group is stopped, or the nominal value if the simulation starts with the circuit breaker closed and the group connected to the network.

This device allows the simulation to be launched smoothly and in a steady state.

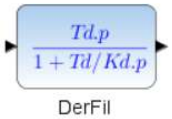
See also
Components with Auto Mode

PSB, DerFil

DerFil

DerFil Filtered derivative with initial condition

Block Screenshot



Description

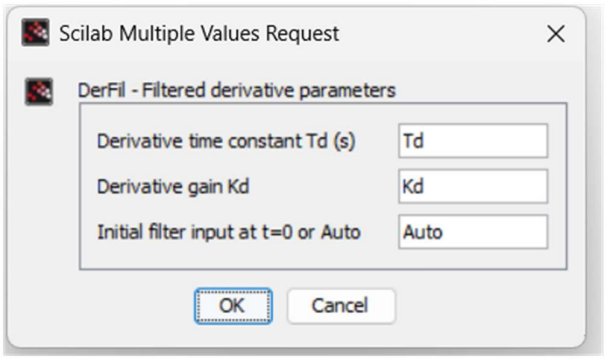
The DerFil block represents a filtered derivative. The time constant Td and the gain Kd of the differentiator are programmable. If known, it is possible to program the initial input of the filter to t=0 to obtain a zero derivative at the start of the simulation.

With the following remark:

$$\frac{T_d.p}{1+\frac{T_d}{K_d}.p} = \frac{K_d.T_d.p}{K_d+T_d.p}$$

, the derived action can be cancelled by programming $K_d = 0$, if necessary.

Parameters



Alternative forms

Interfacing function SegPal\macros\DerFil.sci

Computational function SegPal\src\c\DerFilC.c

Test program SegPal\demos\DerFil.zcos , SegPal\demos\DerFil_Auto.zcos

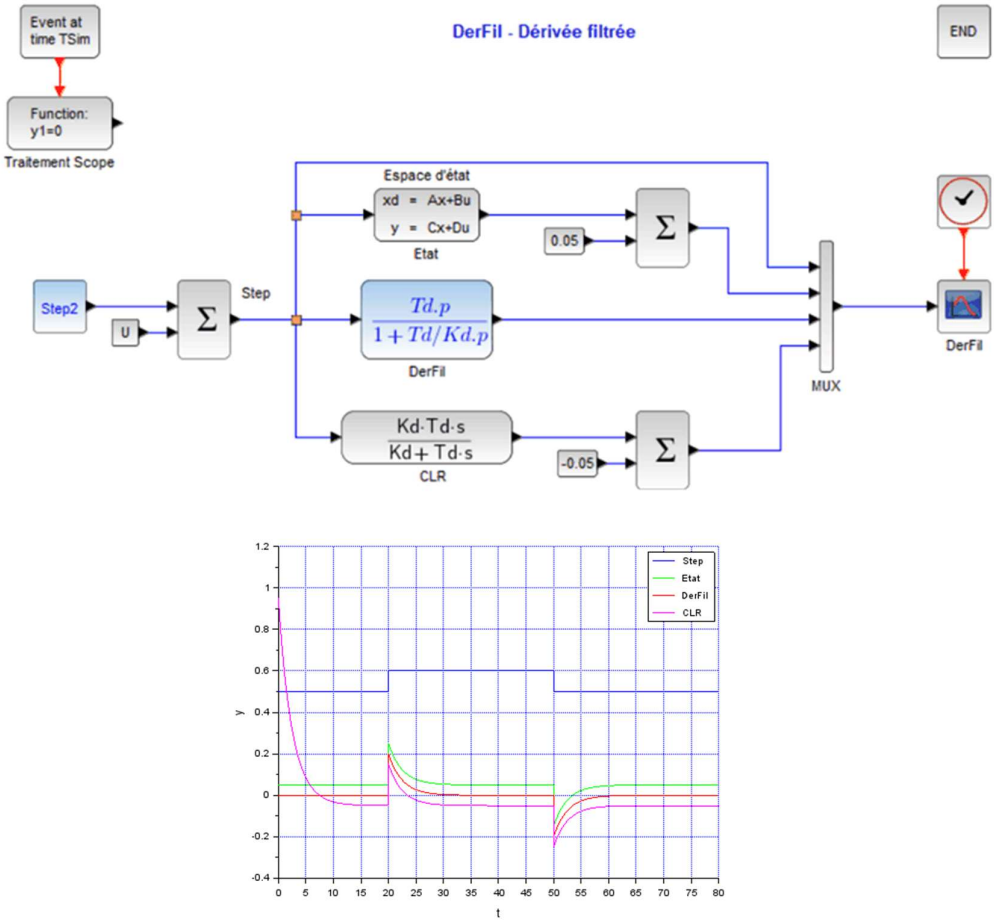


Figure 27 : DerFil.zcos test program

Notes

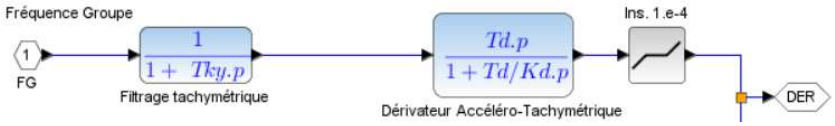


Figure 28 : Accelerometer-tachometer in HydroSim governor module

The DerFil block is for example used in the REGxx models of a speed governor to represent the accelerometer-tachometer derivative, allowing phase advance to be obtained in the adjustment of the regulation loop.
Depending on the circuit breaker status, the initialization value is 0 if the unit is stopped, or the nominal value if the simulation starts with the circuit breaker closed, unit coupled to the network.
This device allows the simulation to be launched smoothly, and in steady state.

The example shown in *Figure 27* illustrates three possible constructions of the same differentiator filter. The state representation in matrix form (A, B, C, D) is calculated in the preamble before the simulation is launched. (***Simulation → Set Context***) :

```
TSim=80  // Temps de simulation

Td=5  // Constante de temps du dérivateur
Kd=2  // Gain du dérivateur

U=0.5  // Valeur initiale du filtre

//-----------------------------------------------------
s=%s;
A=0;B=0;C=0;D=0;X0=0;
if Kd<>0 then
 H=Kd*Td*s/(Kd+Td*s)
 Sys=tf2ss(H) // Transfer to state-space

 A=Sys.A  // Matrice d'état [A,B,C,D]
 B=Sys.B
 C=Sys.C
 D=Sys.D
 X0=-D*U/C // Conditions initiales
end
```

Figure 29 : Context of the DerFil.zcos test program

The state representations and the DerFil derivation filter produce an identical result, with, in particular, an initialization to 0 of the derivative action.
Scilab's standard CLR function does not allow us to obtain this result, hence a non-zero derivative at t=0.

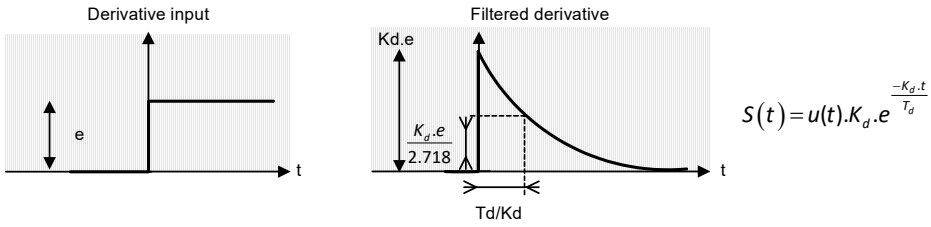


Figure 30 : Reminder of the principle of the filtered derivative

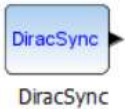
See also Components with Auto Mode

- CLRseg
- PSB

DiracSync

Generation of Dirac comb synchronized to a start date

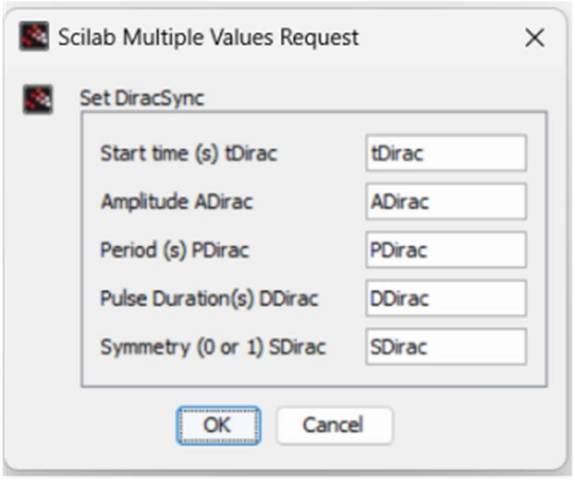
Block Screenshot



Description

The DiracSync block allows the generation of a Dirac comb, generally used as a speed signal disruptor, to test or quantify the performance of the speed regulation.

Parameters

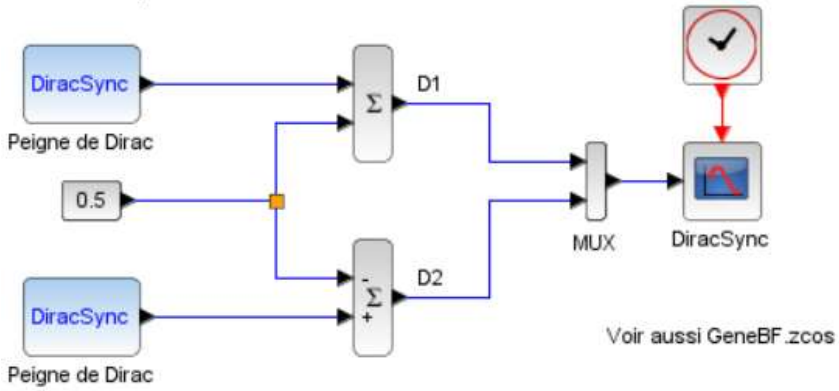


Alternative forms -

Interfacing function SegPal\macros\DiracSync.sci

Computational function SegPal\scr\c\DiracSyncC.c

Test program SegPal\demos\DiracSync.zcos



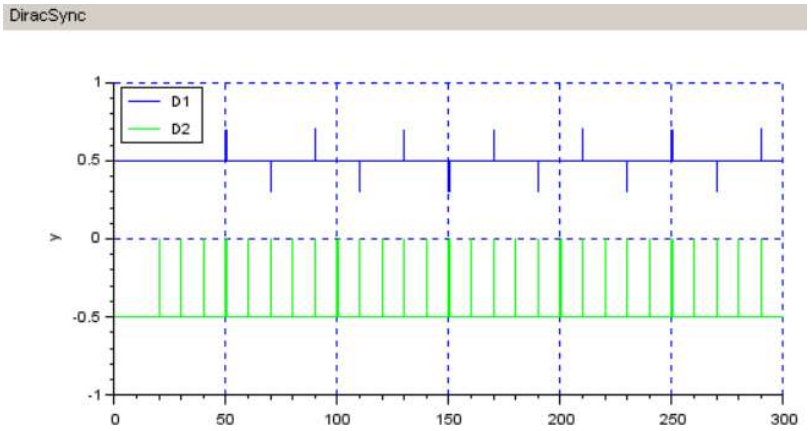


Figure 31 : Test program DiracSync.zcos

Notes

The "Symmetry" option allows the direction of the Dirac delta function to be reversed at each sampling period (channel D1 in the graph above).

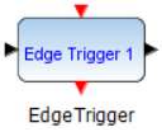
The DiracSync block is like the PULSE block  , standard in the XCOS Sources Palette.

The interface is simplified here and adapted to the **HydroSim** project.

EdgeTrigger

Edge detector of an input square signal

Block Screenshot



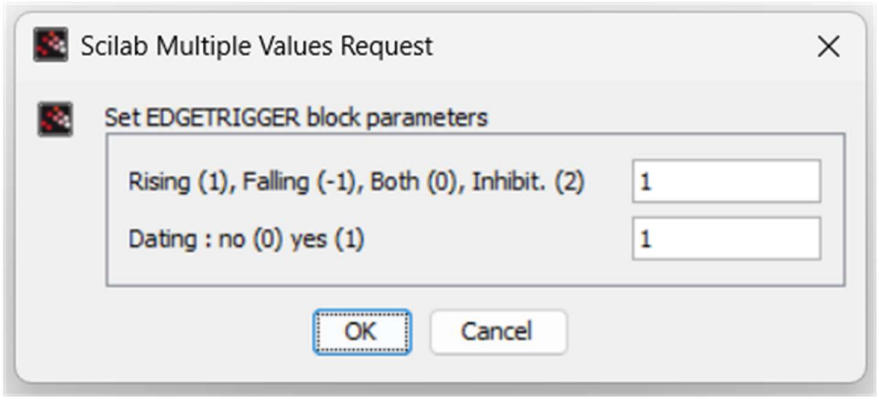
Description

The EdgeTrigger block allows the counting of edges detected on an input square signal, and provides the output dating of the recorded events.

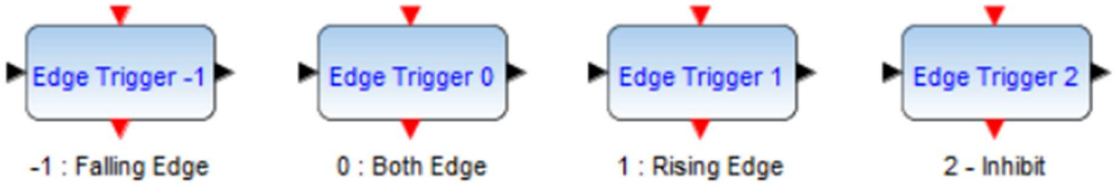
It also provides a resynchronization signal at the appearance of each event.

The function counts, depending on the programmed option, the rising or falling edges of the input signal, or all edges.

Parameters



Alternative forms

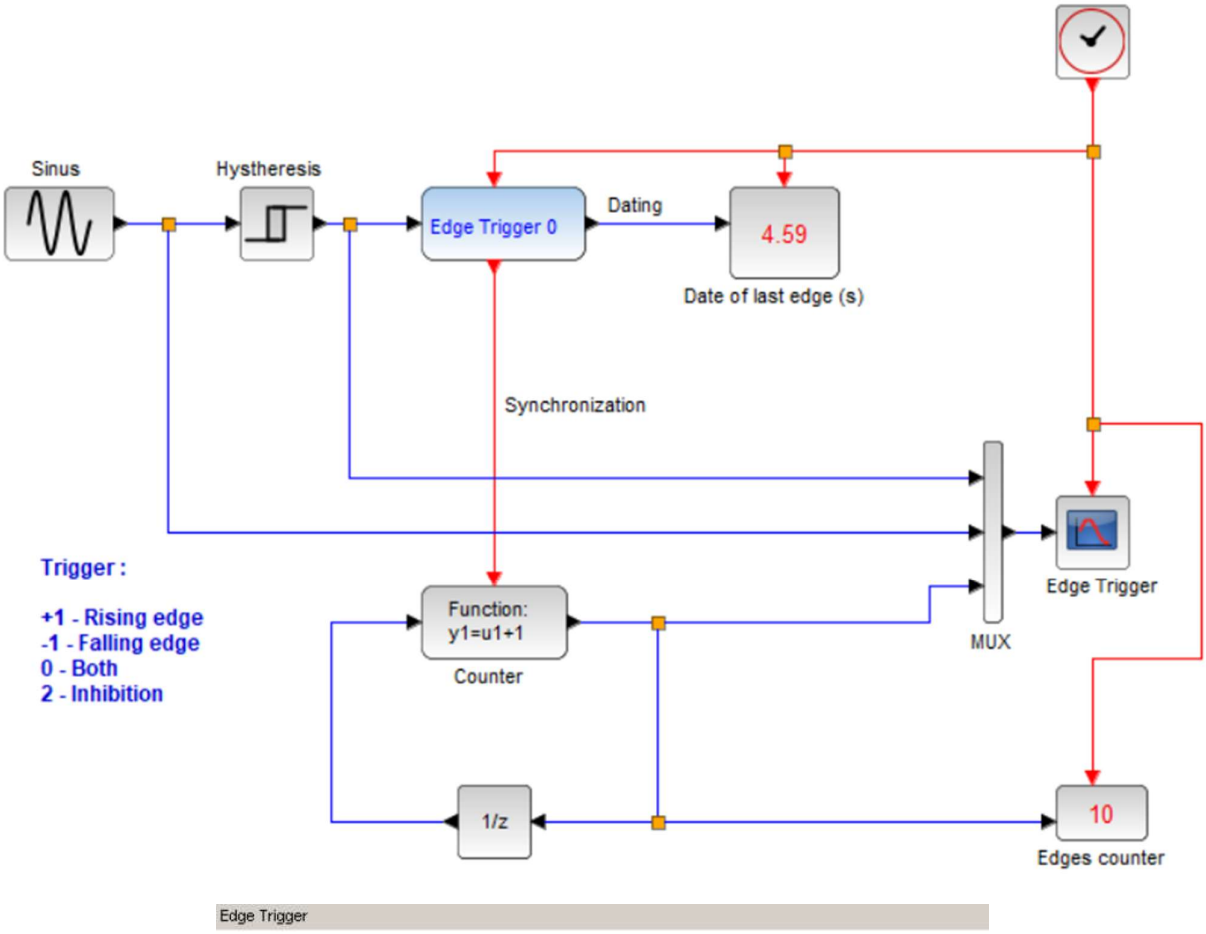


-

Interfacing function SegPal\macros\EdgeTrigger.sci

Computational function SegPal\scr\c\EdgeTriggerC.c

Test program SegPal\demos\EdgeTrigger.zcos



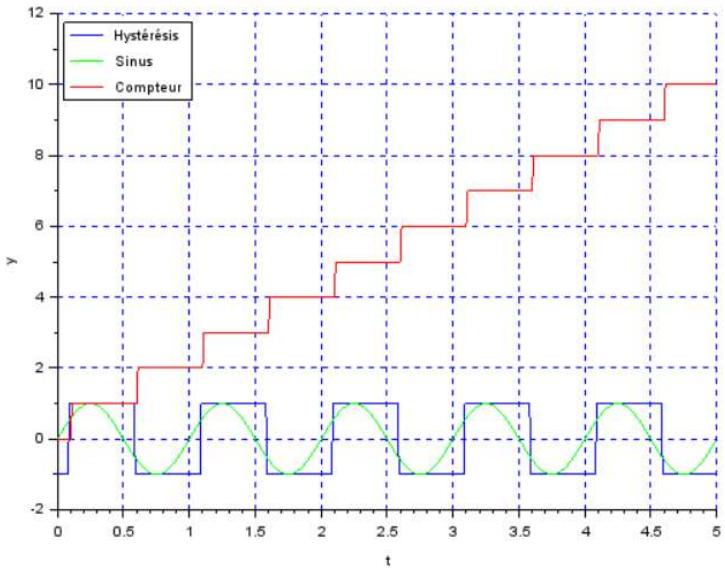


Figure 32 : EdgeTrigger.zcos test program

Notes An application of the EdgeTrigger block function is also shown in the Busy.zcos test program.

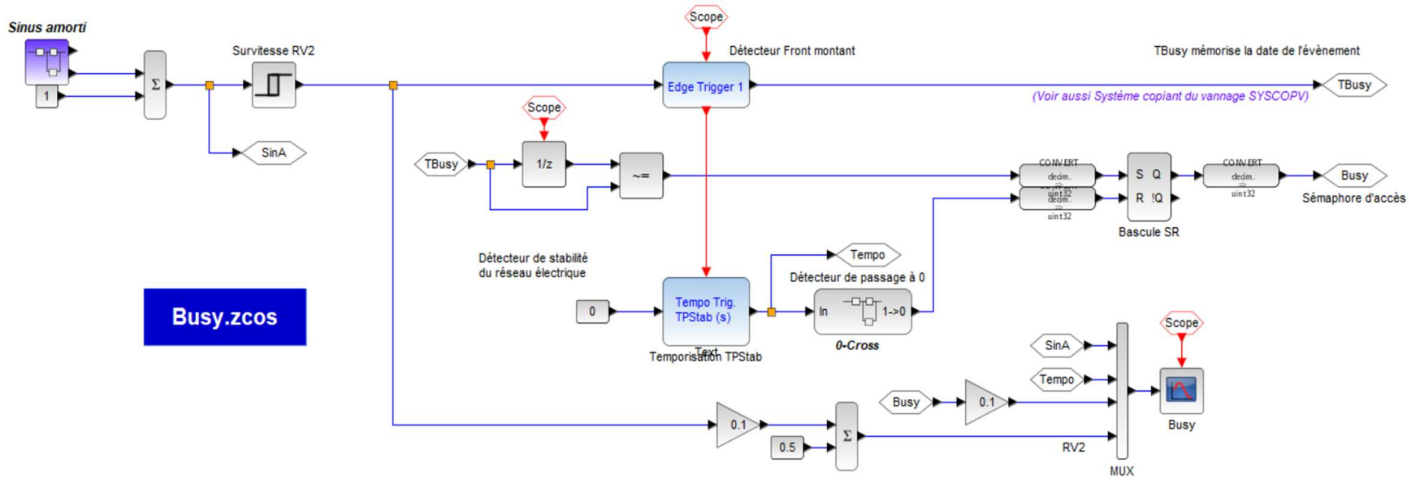


Figure 33 : Busy.zcos test program (1/2)

The "**Busy.zcos**" test program represents a pre-study allowing the detection of instability on the network frequency, following a disturbance of the speed signal, itself generated by a load step on the network.

n this program, a damped sinusoid **sinus(x)/x** is generated at the input. During the transient, a "Busy" RS flip-flop is armed as long as oscillations persist in a programmable band.

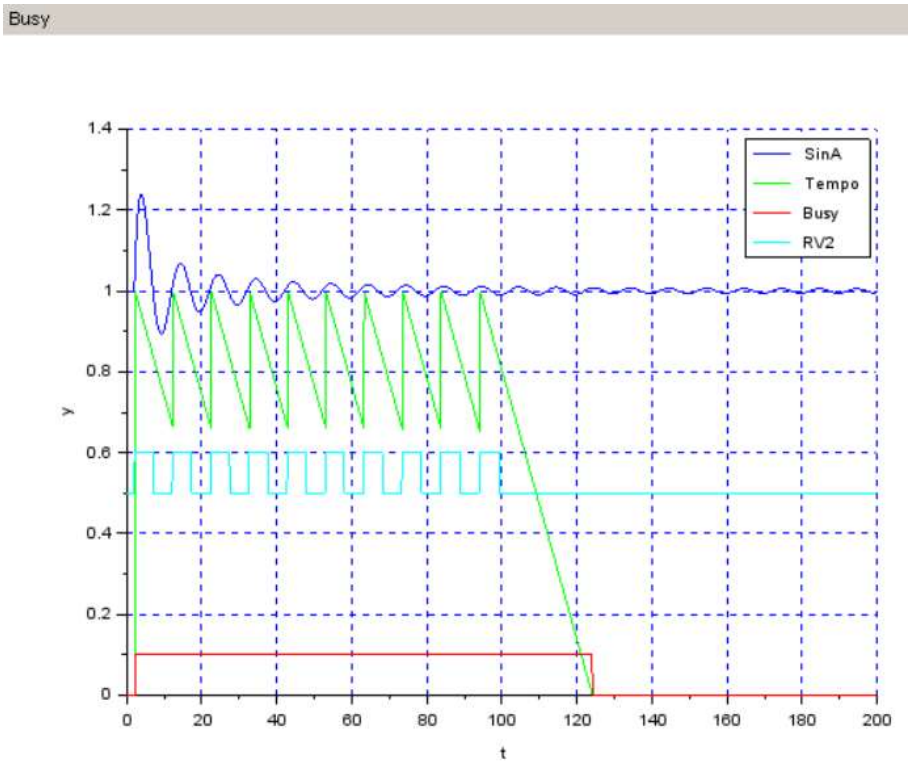


Figure 34 : Test program Busy.zcos (2/2)

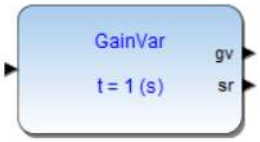
The RS flip-flop is reset to 0 when the network is again detected stable, with the programmable tolerance.

☞ **See also TempoTrig, Sync1 Resynchronization Relay**

GainVar

Jets dispatcher - Switching ramp applied to the position setpoint of an injector for a multi-jets PELTON machine.

Block Screenshot

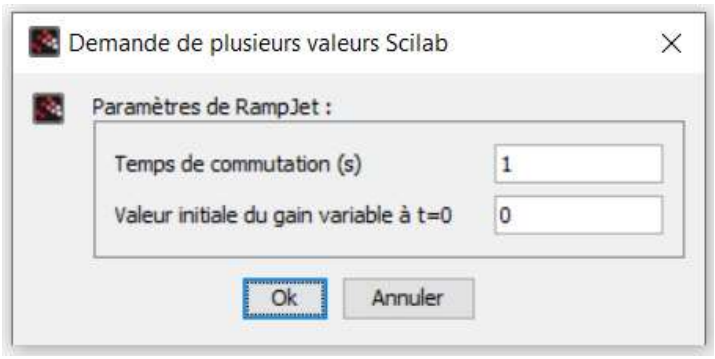


Description

The GainVar component is dedicated to **PELTON vertical axis turbines**, when the programmable option RepOnOff=1 engages the dispatching function. Otherwise, the jet dispatcher is inoperative.

The principles of the jet distributor and its operation are described in document *[10], (Jet distributor for PELTON machines page 73/240*), as well as in the document "**HydroSim - Programming guide**"

Parameters



Alternative forms

-

Interfacing function

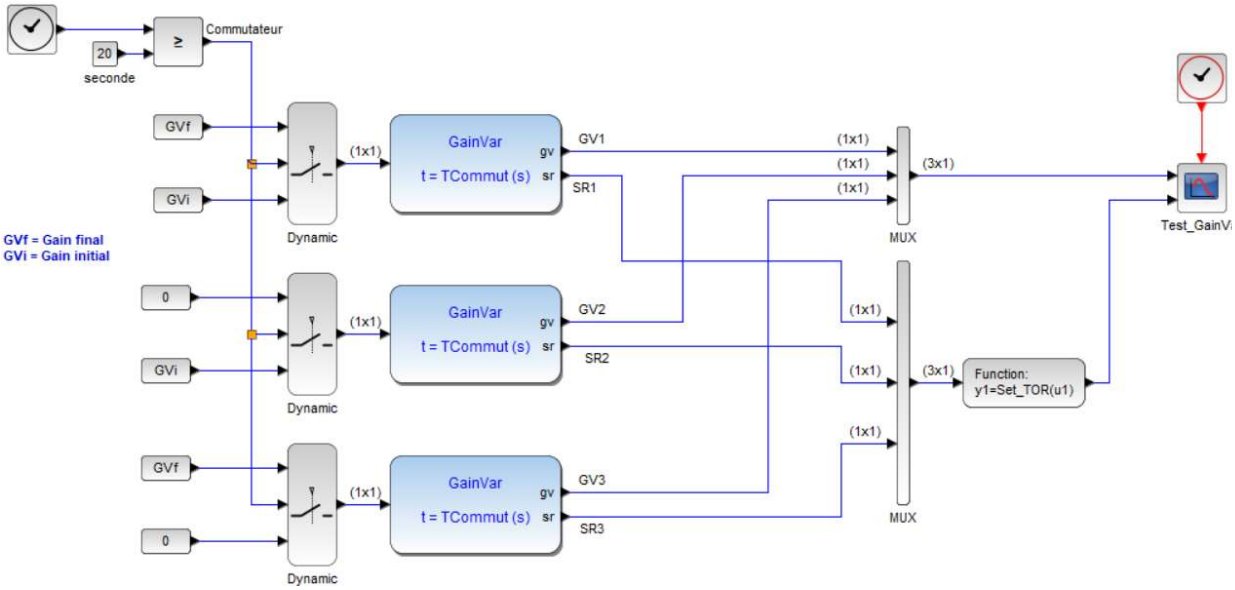
SegPal\macros\GainVar.sci

Computational function

SegPal\scr\c\GainVarC.c

Test program

SegPal\demos\GainVar.zcos et GainVar_V.zcos



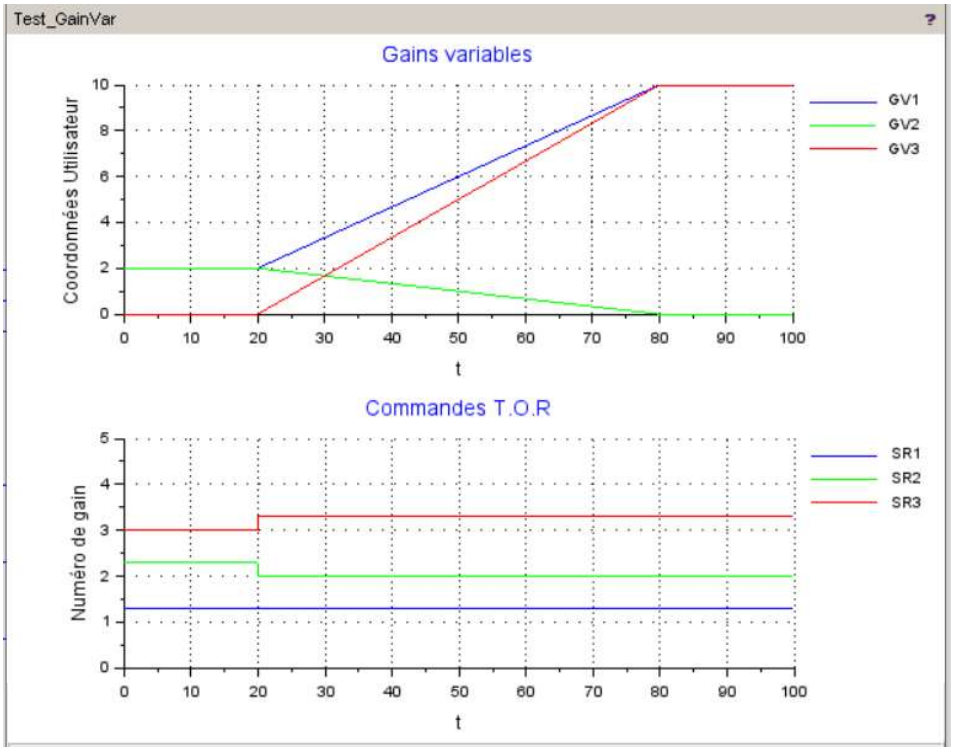
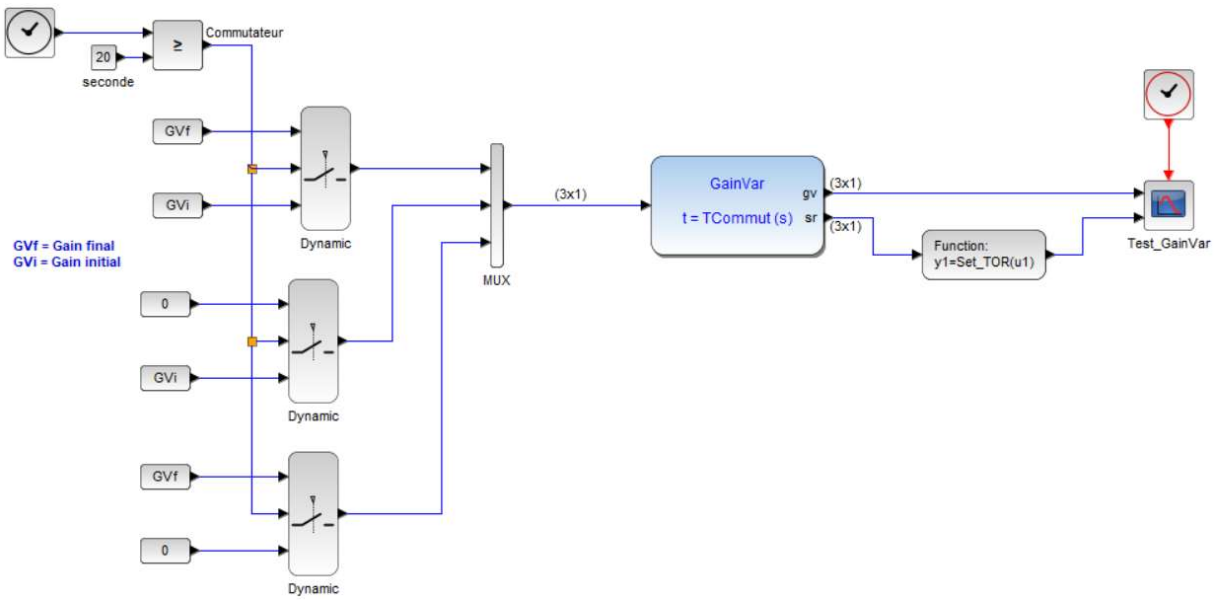


Figure 35 : GainVar.zcos test program – Monovariable Implantation



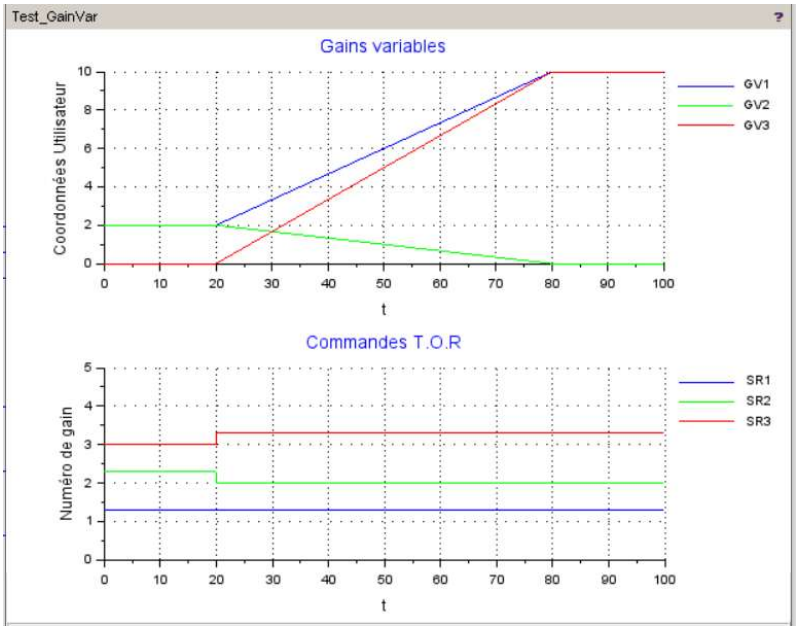


Figure 36 : GainVar_V.zcos test program – Multivariable implementation

Notes

When switching jets, the objective of the GainVar component is to adapt and guarantee the time constants of the injector operating laws, so that they open or close in an identical time, with a constant flow (or power) constraint.

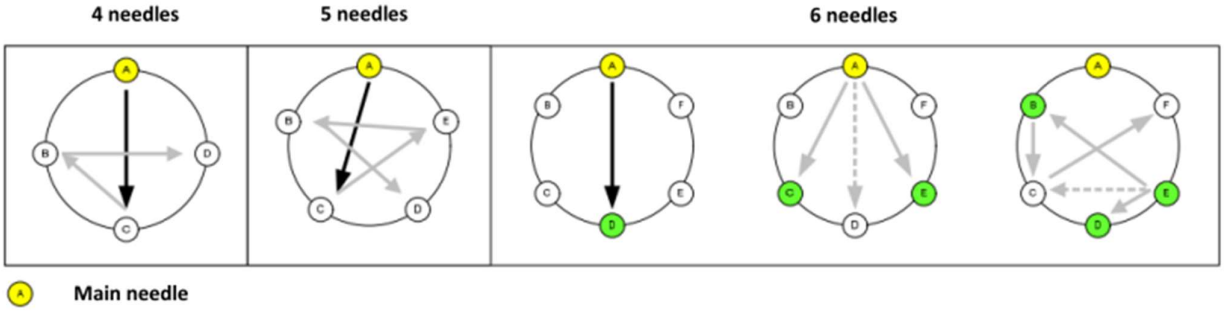


Figure 37 : Principle of jet switching, with respect to the calculations of radial forces on the PELTON wheel

In the HydroSim diagrams of PELTON turbines, the variable gain modulates the injector setpoint in the SYSCOP copying systems of the positioners.

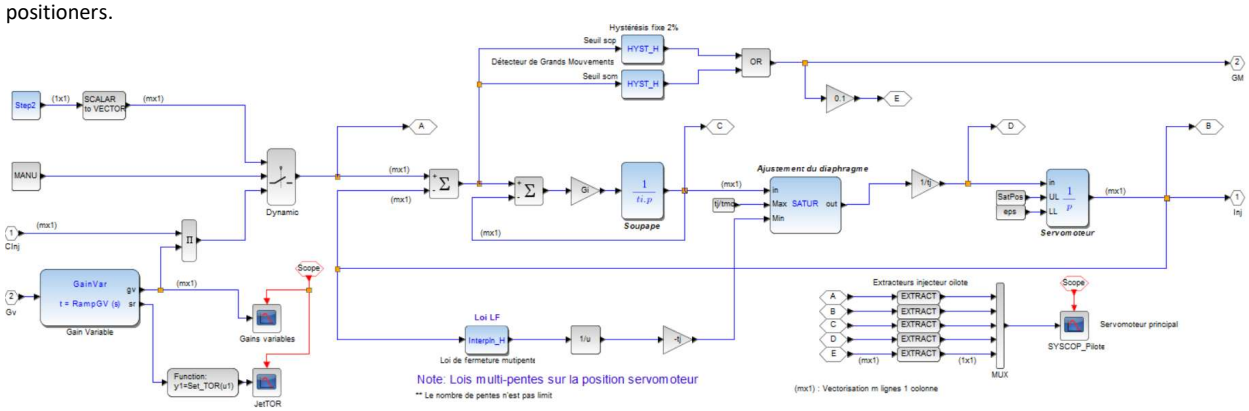


Figure 38 : Extract from systems copying SYSCOP_I of injectors (see PELTON diagram library) – Multivariable implementation

GeneBF — Low Frequency Signal Generator – Disturber

Block Screenshot



Description

GeneBF represents the equivalent of a SuperBlock using the STEP_FUNCTION (Xcos Scilab), DiracSyn, and SinuSync components.

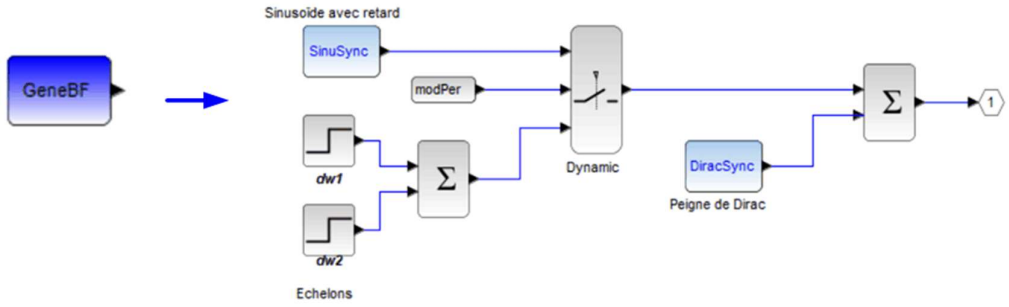


Figure 39 : Equivalence of the GeneBF generator with a SuperBlock

The generator (or disruptor) is used in the "Network" folios of the HYDROTUR simulations, to generate disturbances:

- On the unit speed signal
- On the load setpoint of the unit

Parameters

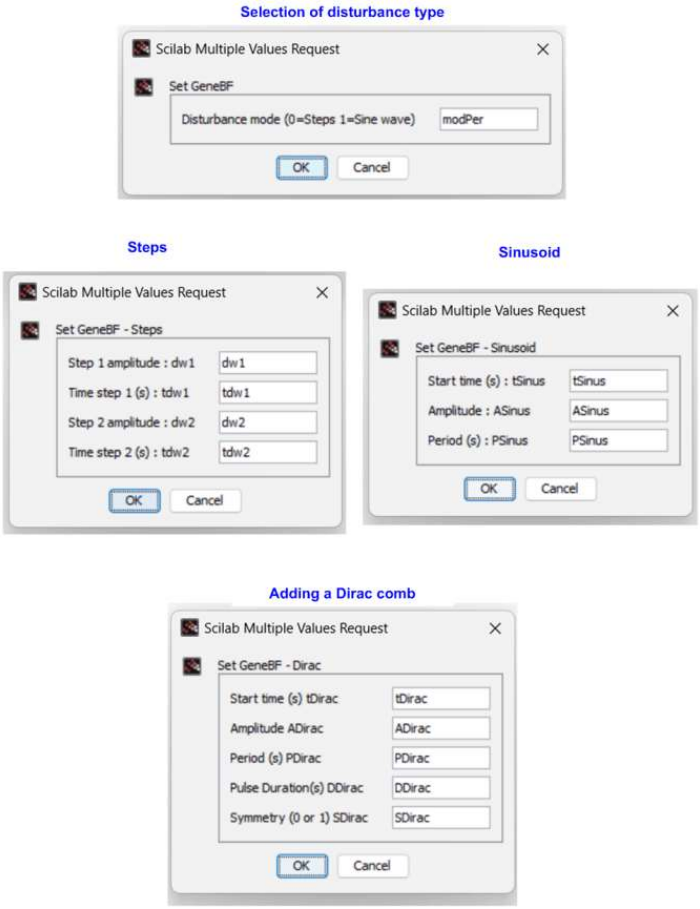


Figure 40 : GeneBF Generator Operator Interface

☞ The reader will also consult:

- document « **HydroSim - Programming** » IN «HydroSim\docs\ » directory

Alternative forms -

Interfacing function SegPal\macros\GeneBF.sci

Computational function SegPal\src\c\GeneBFC.c

Test program SegPal\demos \GeneBF.zcos

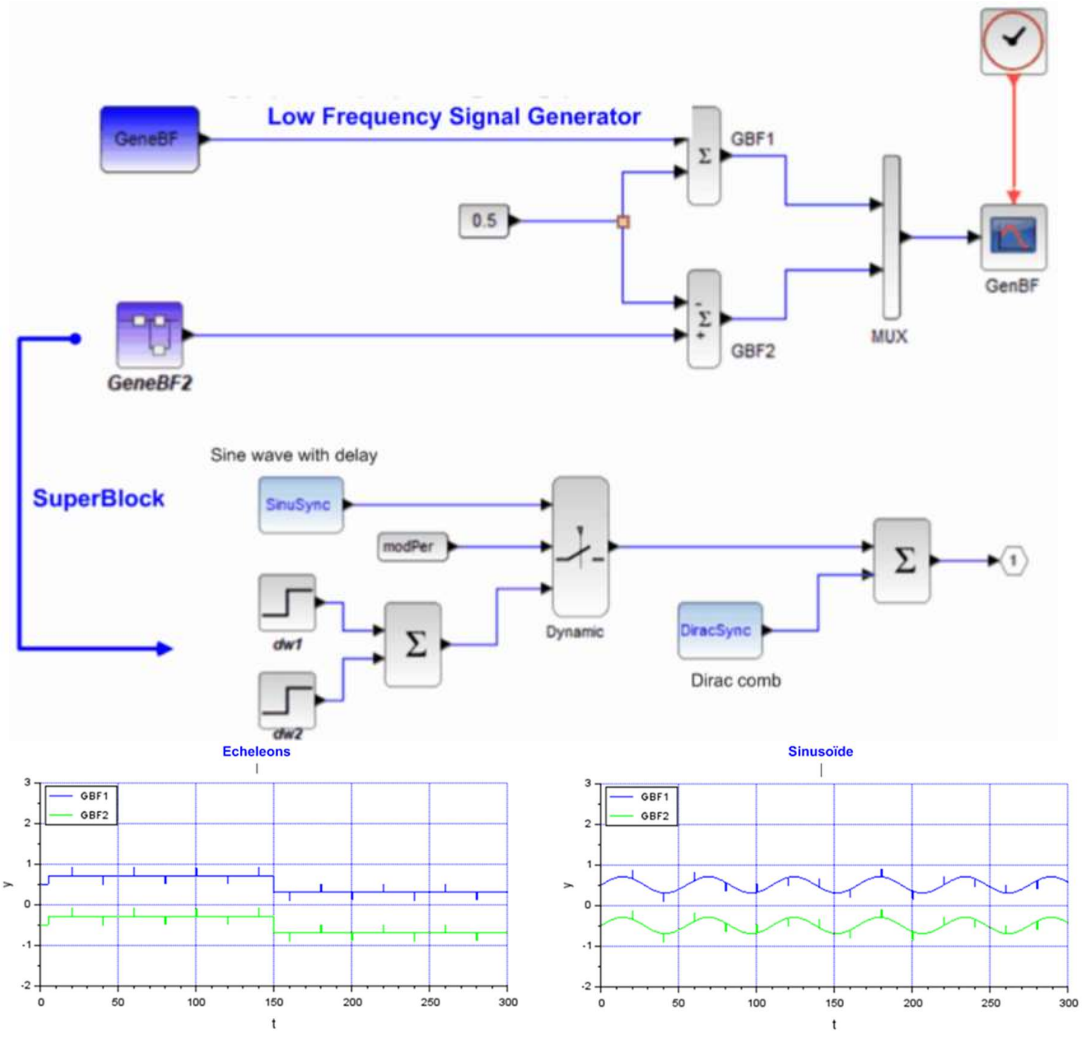


Figure 41 : GeneBF.zcos Generator test program

Notes Use of the GeneBF disruptor in the "Reseau" modules of **HydroSim\TURBINES\ simulators** directories

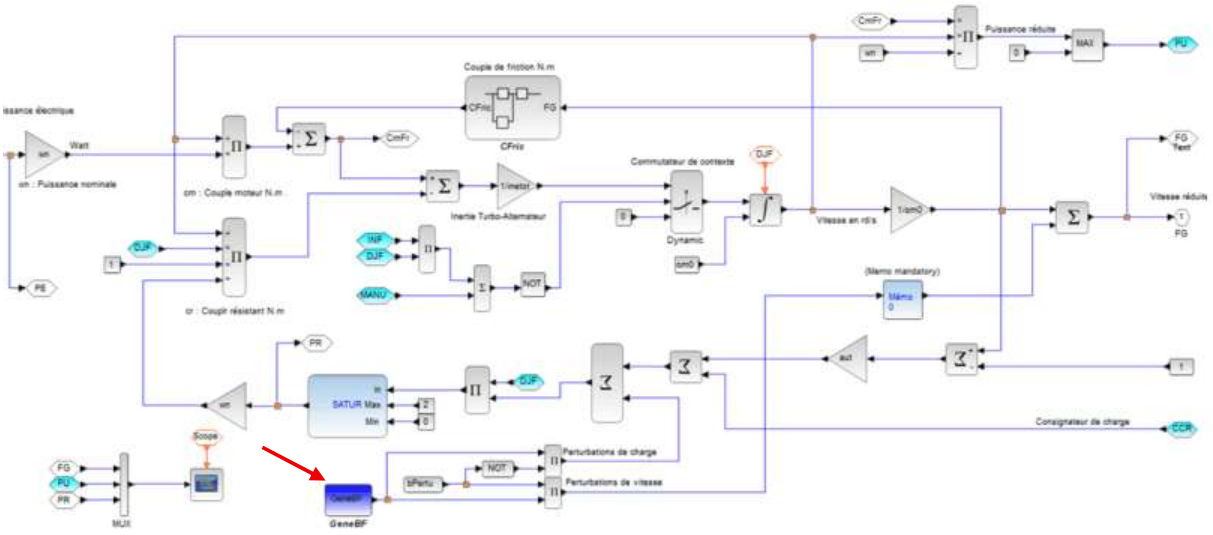
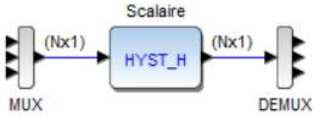


Figure 42 : Use of the GeneBF disruptor in the “Reseau” modules of HydroSim simulators

Hysteresis_H

Multivariate hysteresis

Block Screenshot



Description

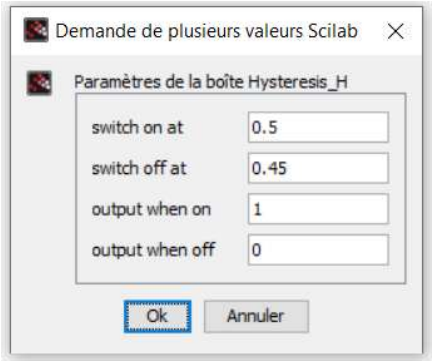
The **Hysteresis_H** component allows its outputs to switch between two specified scalar values.

When the relays are closed, they remain closed until the corresponding inputs fall below the value of the "*Switch off at*" parameter.

When the relays are open, they remain open until the corresponding inputs exceed the value of the parameter "*Switch on at*".

The block accepts a multivariable input Nx1 and generates an output of type Nx1 as well.

Parameters



Alternative forms -

Interfacing function SegPal\macros\ Hysteresis_H.sci

Computational function SegPal\scr\c\ Hysteresis_HC.c

Test program SegPal\demos\ Hysteresis_H.zcos

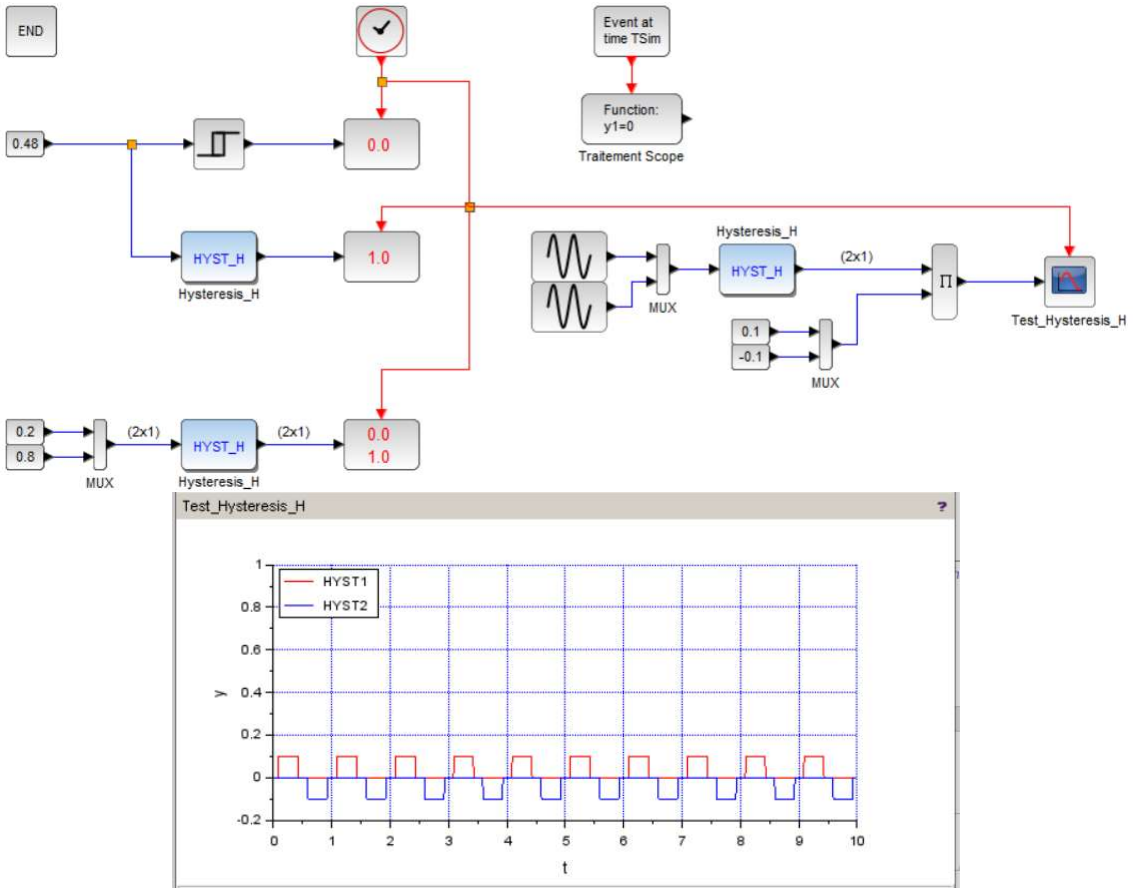


Figure 43 : Hysteresis_H.zcos test program

See also

Figure **18**
Relais_H

InterpIn_H

Multivariable linear interpolation

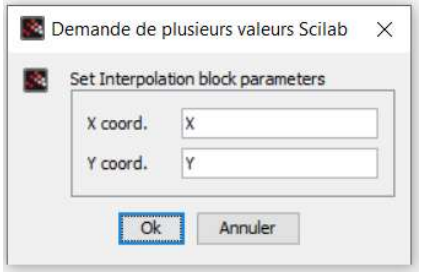
Block Screenshot



Description

Let XY be a space describing a set of points in the plane, with increasing abscissas, and X a set of abscissas, this function returns in the output vector S the corresponding ordinates, calculated by linear interpolation of the input data U.

Parameters



Vectors X and Y can be either 2 row vectors or 2 column vectors of the same dimension.

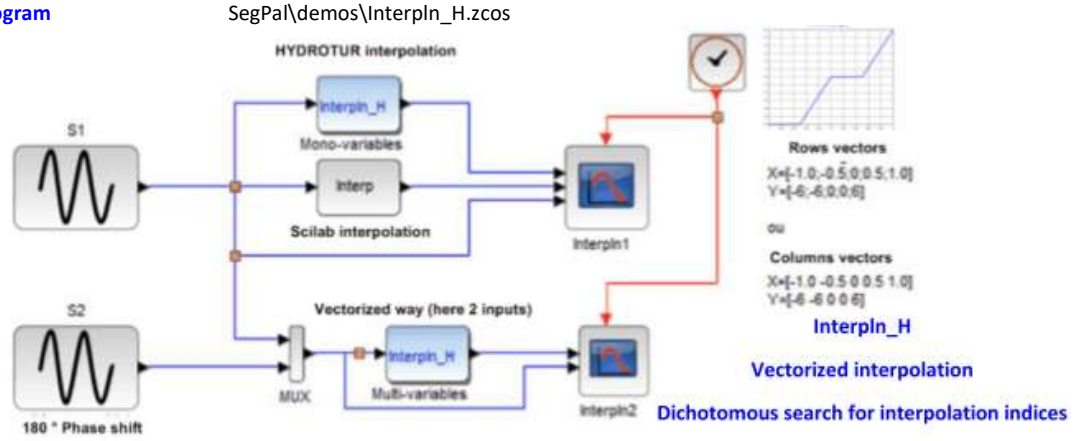
Reminder: the abscissas X must be strictly increasing.

Alternative forms -

Interfacing function SegPal\macros\Interpln_H.sci

Computational function SegPal\scr\c\Interpln_HC.c

Test program SegPal\demos\Interpln_H.zcos



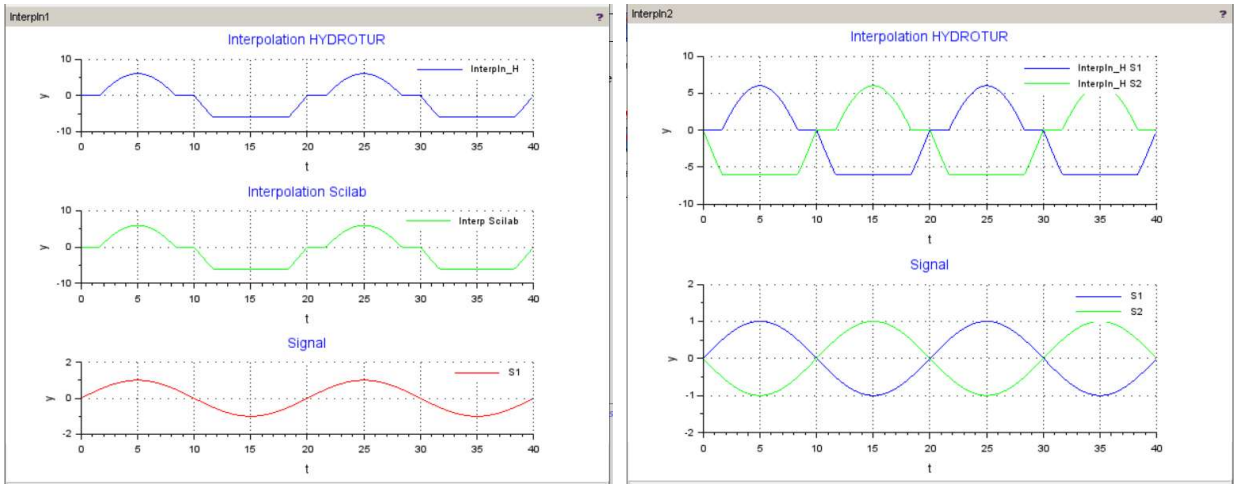
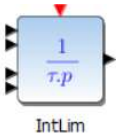


Figure 44 : Interpln_H.zcos test program

IntLim

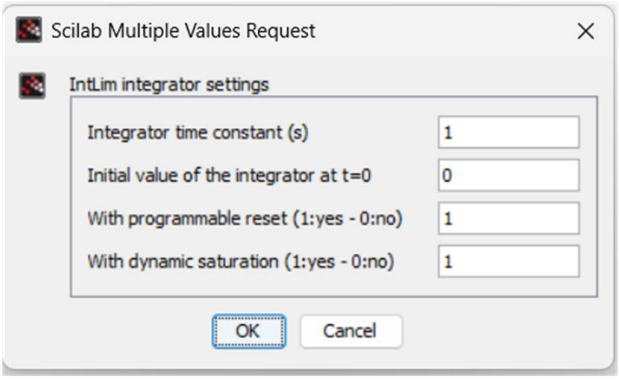
Pure integrator with reset and external saturations

Block Screenshot



Description The IntLim block integrates its input signal. It includes, as an option, saturation control devices or event reset.
The initial value of the block is also programmable.

Parameters



Alternative forms



Interfacing function SegPal\macros\IntLim.sci

Computational function SegPal\scr\c\IntLimC.c

Test program SegPal\demos\IntLim.zcos

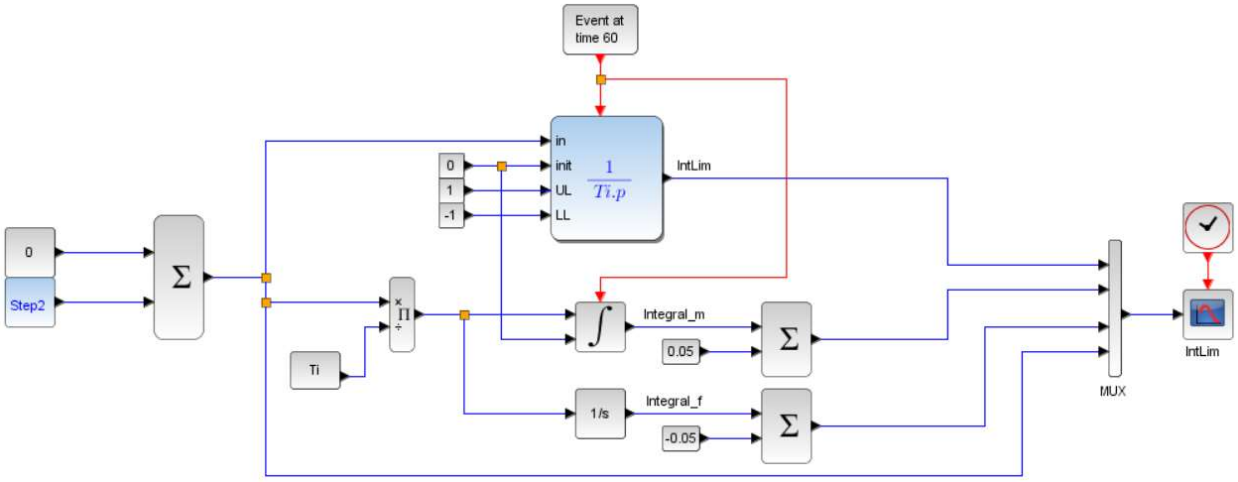


Figure 45 : Comparison of the integrators IntLim (**SepPal**) and Integral_m, Integral_f (**Scilab-Xcos**)

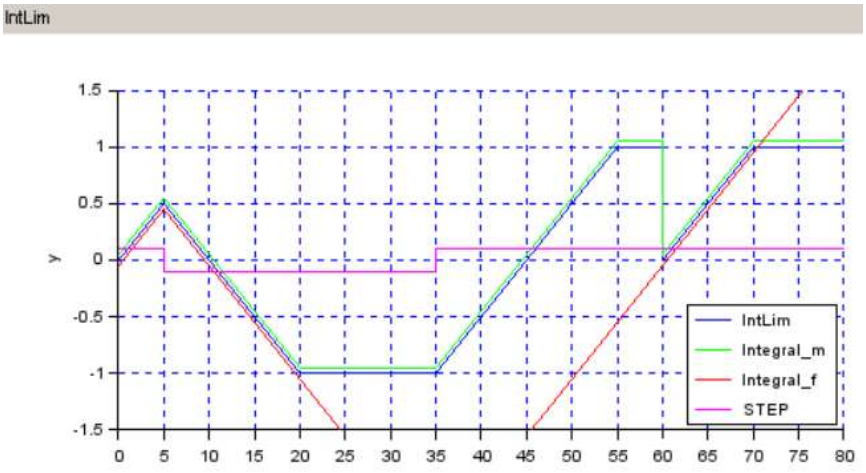


Figure 46 : IntLim.zcost test program

Notes The IntLim integrator is primarily used in the REGxx PID blocks of **HydroSim** speed controllers.

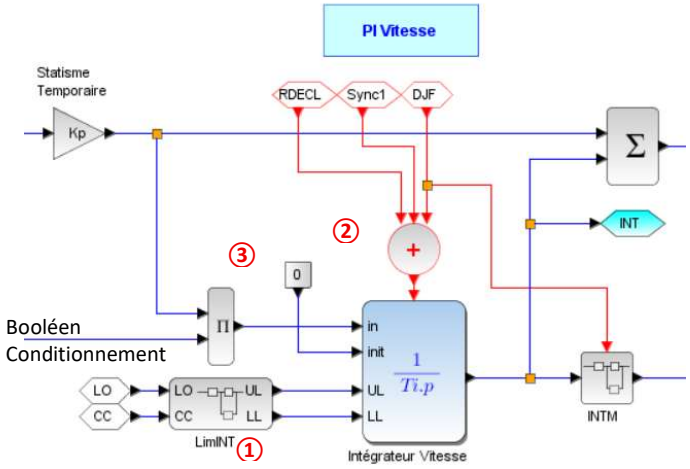


Figure 47 : Integrator IntLim in the **HydroSim** speed governor

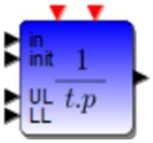
This version includes:

- An external saturation device, variable depending on the load imposed on the turbo-alternator group (LimINT sub-block) ①
- A reset mechanism upon receipt of an event: ② ③
  - ✓ **RDECL** : Group trigger relay
  - ✓ **DJF**: Group circuit breaker closure
  - ✓ **Sync1**: Resynchronization due to significant speed difference

IntSPE

Special integrator with reset, external saturations, and fast desaturation

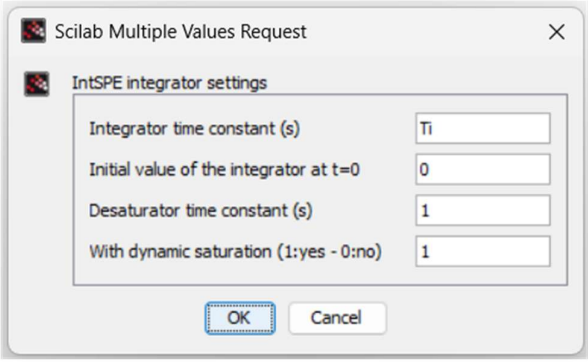
Block Screenshot



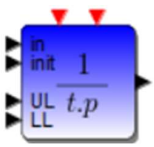
Description

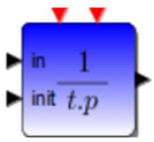
The IntSPE block integrates its input signal. It includes, as an option, saturation control devices, or event reset.
The initial value of the block is also programmable.
The block differs from the "**IntLim**" block by the possibility of event-driven desaturation. The desaturation time (reset to 0) is programmable.
When the "Desaturation" event occurs, **InsSPE** stores the value of the integrator and makes it tend towards 0 in the programmed time.

Parameters



Alternative forms





Interfacing function

SegPal\macros\IntSPE.sci

Computational function SegPal\scr\c\IntSPEC.c

Test program SegPal\demos\IntSPE.-zcos

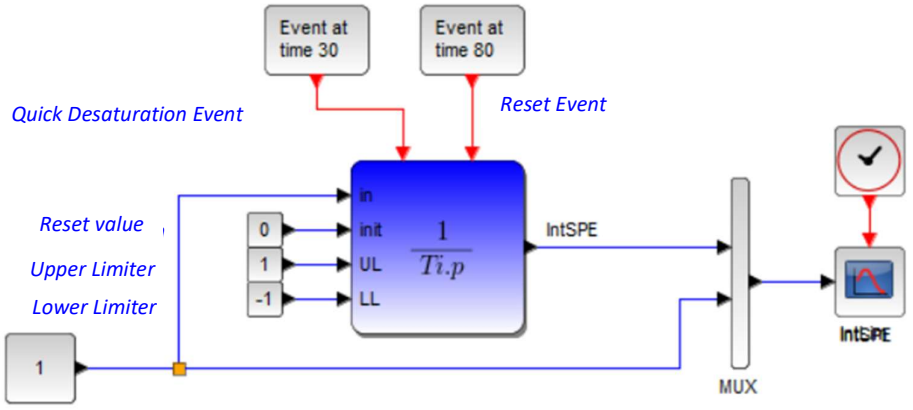


Figure 48 : IntSPE Leak integrator

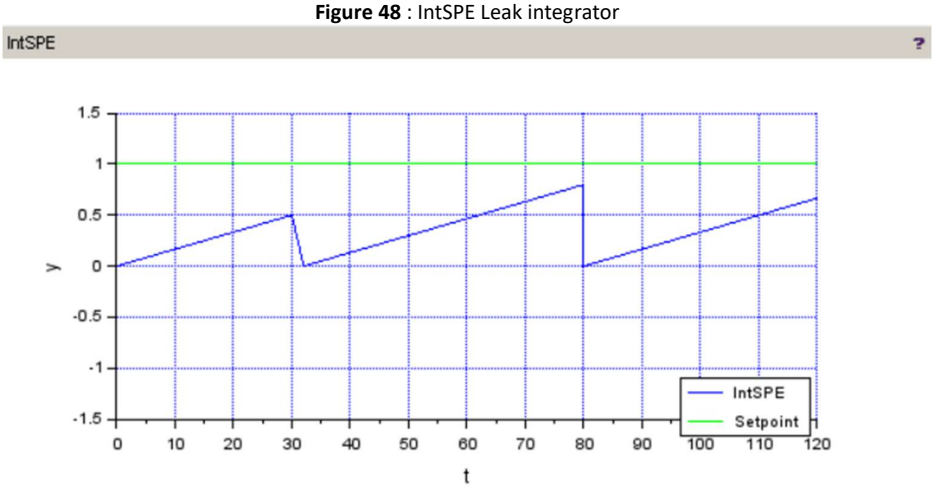


Figure 49 : IntSPE.zcos test program

Notes The time constant of the desaturation cannot be less than 0.1 s.

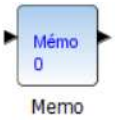


Figure 50 : IntSPE Integrator Desaturation Time Control

Memo

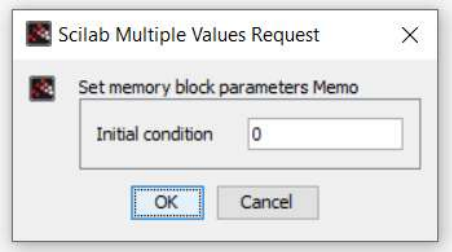
Pure delay

Block Screenshot



Description The Memo block delays its input by one solver sampling period.
The initial value(s) of the block are programmable, and displayed in the XCOS component

Parameters



Alternative forms -

Interfacing function SegPal\macros\Memo.sci

Computational function SegPal\scr\c\MemoC.c

Test program SegPal\demos\Memo.zcos

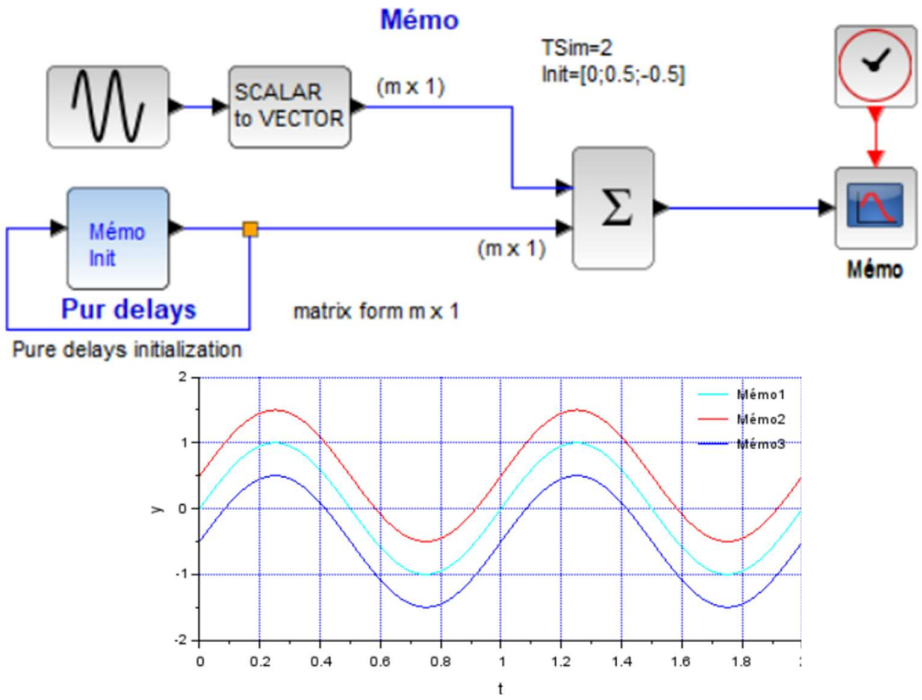
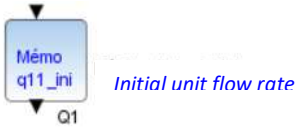


Figure 51 : Memo.zcos test program

Notes



The Memo block is used in the TURBADDUC block of the FR5 model (**HydroSim**), to initialize the initial flow rate when launching the simulation at t=0 s.
This device allows the simulation to be launched smoothly and in steady state.

Pow

Raising a positive number to the power

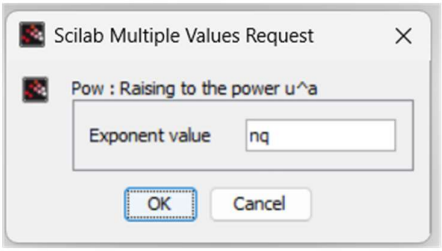
Block Screenshot



Description

The Pow block raises its positive input "u" to the power of "a".
The numeric or symbolic value of the exponent is displayed in the block interface.

Parameters

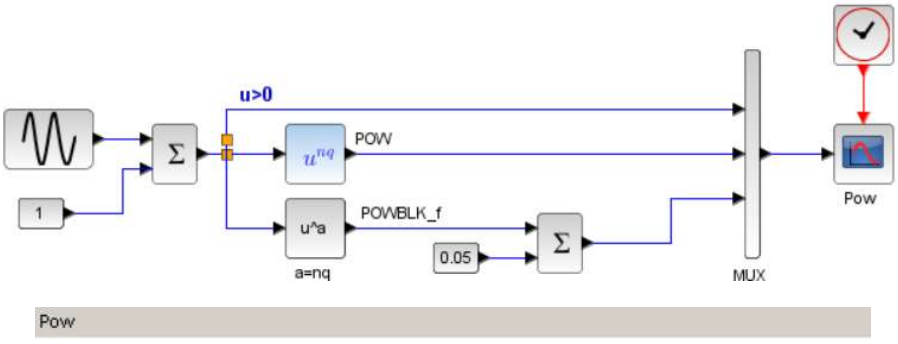


Alternative forms -

Interfacing function SegPal\macros\ Pow.sci

Computational function SegPal\scr\c\ ExponentC.c

Test program SegPal\demos\ Pow.zcos - Pow2.zcos



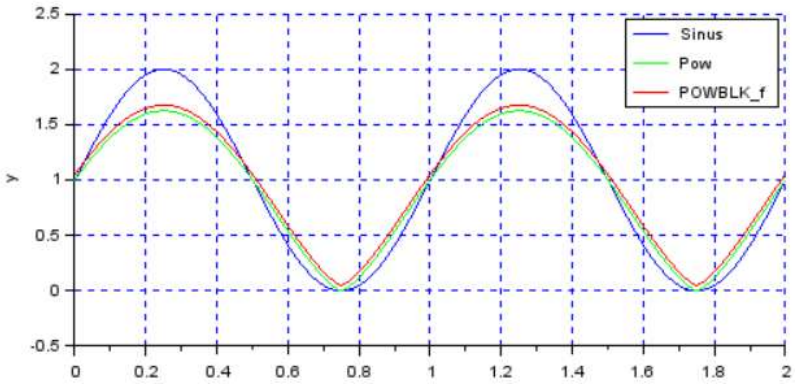


Figure 52 : Pow.zcos test program

Notes The Pow block is used in the TURBADDUC block of the FR2 or PELTON (**HydroSim**) models to give an image of the Flow/Opening relationship with an approximation factor nq=0.7

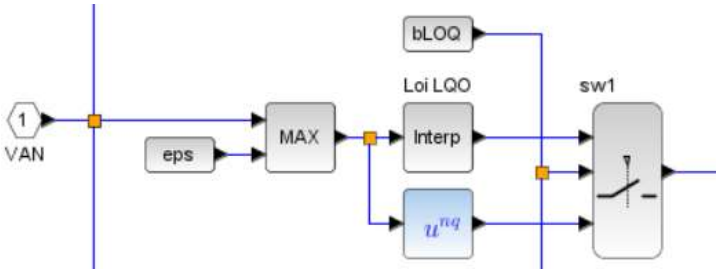
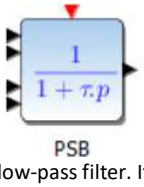


Figure 53 : Approximation of the Flow/Opening law in TURBADDUC model FR2 (**HydroSim**)

PSB

1st order low-pass filter with reset and external saturations

Block Screenshot

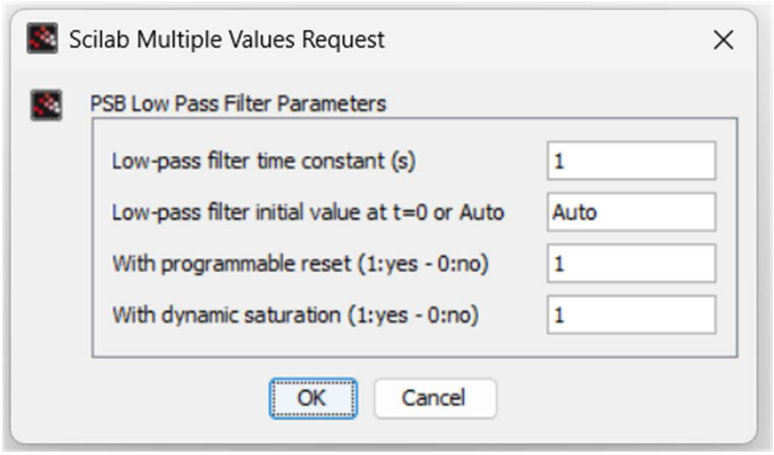


Description

The PSB block represents a first-order low-pass filter. It includes, as an option, saturation control devices or event reset.
The initial value of the block is also programmable.

- In normal mode, the PSB filter is initialized according to the initial value indicated in its function box.
- In automatic "Auto" mode, the filter is automatically initialized according to its input value and maintains its output equal to its input value.

Parameters



Alternative forms



-

Interfacing function

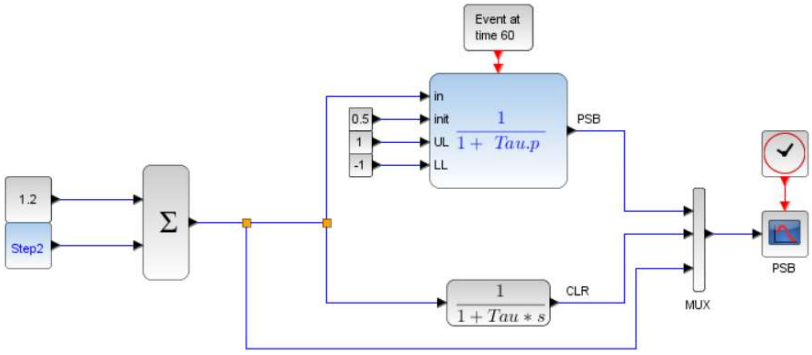
SegPal\macros\ PSB.sci

Computational function

SegPal\scr\c\ PSBC.c

Test program

SegPal\demos\ PSB.zcos



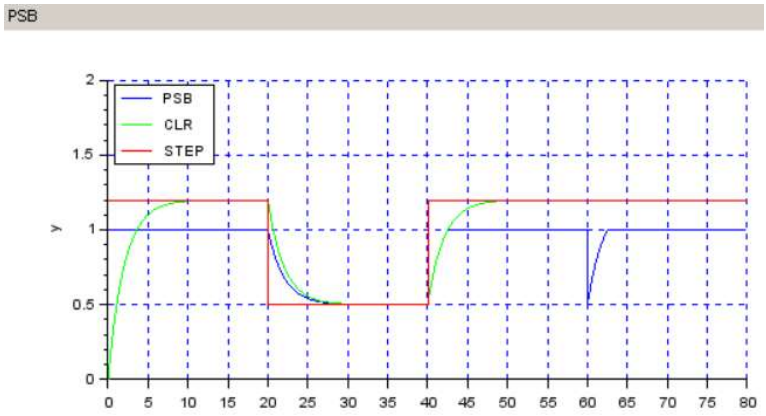


Figure 54 : PSB.zcos test program

Notes

The PSB component improves upon its CLR equivalent insofar as PSB starts at a steady filter operating temperature.

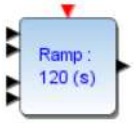
See also Components with Auto Mode

- CLRseg
- DerFil

Ramp

Integrator ramp with reset and external saturations

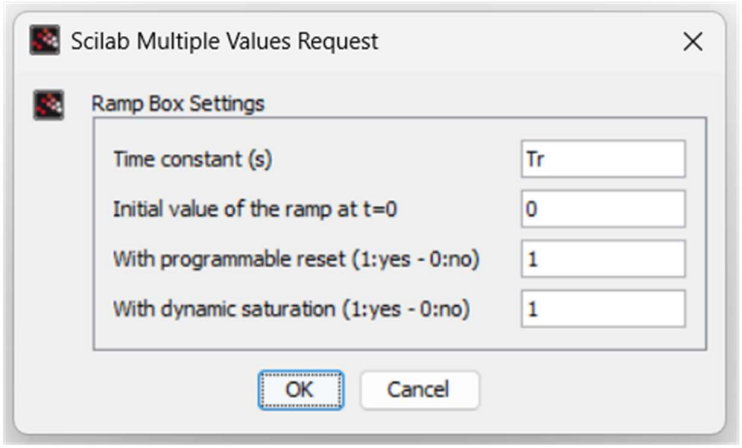
Block Screenshot



Description

The Ramp component generates an incremental signal, parameterized by its time constant, which is the time defined to increase the output signal from 0% to 100% of its value.

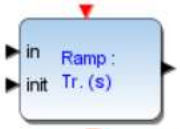
Parameters



Alternative forms









-

Interfacing function SegPal\macros\ Rampe.sci

Computational function SegPal\scr\c\ RampeC.c

Test program SegPal\demos\ Rampe.zcos

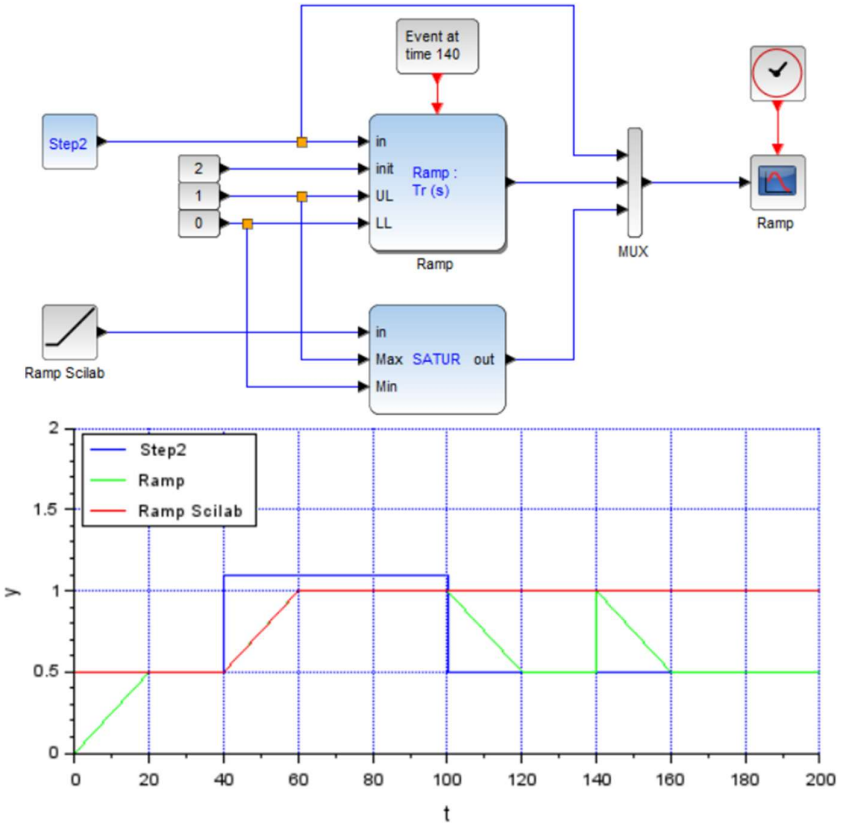


Figure 55 : Rampe.zcos test program

Notes

Figure 56 illustrates the use of the Ramp component in the load loggers of the **HydroSim** models.
This is accessible from the Speed Regulator block (REGxx), then in the CC load module.

The Ramp block output is limited to 0 by lower value. The upper value is a function of the LO opening limitation imposed on the group (see **HydroSim** : LO opening limiter module).

The “in” input of the load setpoint in the Ramp block is zero if the group circuit breaker is open (DJF=0).

At the time the circuit breaker closes (event input DJF, circuit breaker closing date – parameter TDJF), the Ramp block is reset according to the value of the TDJF time:

- If TDJF=0, the simulation is launched with the unit on the network (DJini=1). In this case, the ramp directly equals its setpoint. This device is specially designed to avoid generating a transient regime when launching the simulator.

- If TJF > 0 (DJini=0), the ramp will apply to circuit breaker closing, from the zero value, up to the setpoint value.

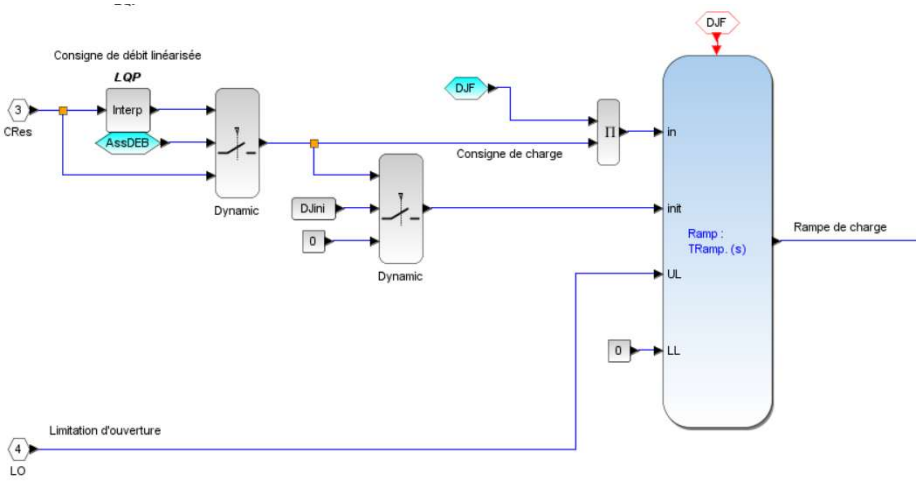


Figure 56 : Extract from the load recorder in **HyDroSim** models

Relais_H

Multiple relays in parallel on a 1x1 input signal

Block Screenshot



Description

This component allows the management of N relays connected in parallel on a monopolar input signal. It therefore provides an output (Nx1) representing the states of all of these relays, according to the following equivalent diagram:

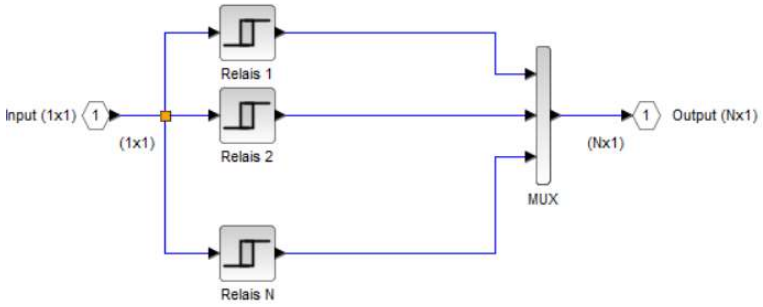
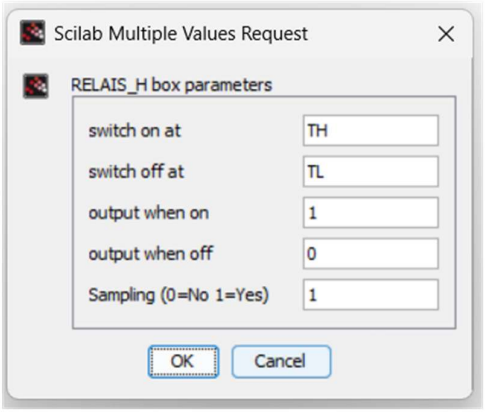


Figure 57 : Principle of the Relay_H component

Parameters



- TH : Matrix (Nx1) representing a table of high state switching thresholds
- TL : Matrix (Nx1) representing a table of low state switching thresholds

Alternative forms



The discretization of the component, in its sampled form, makes it possible to improve the response time of the global simulation of a simulated process, when the variation of the RELAIS_H input is relatively slow.

Interfacing function SegPal\macros\ Relais_H.sci

Computational function SegPal\scr\c\ Relais_HC.c

Test program SegPal\demos\ Relais_H.zcos, Répartiteur_Jets1.zcos, Répartiteur_Jets2.zcos

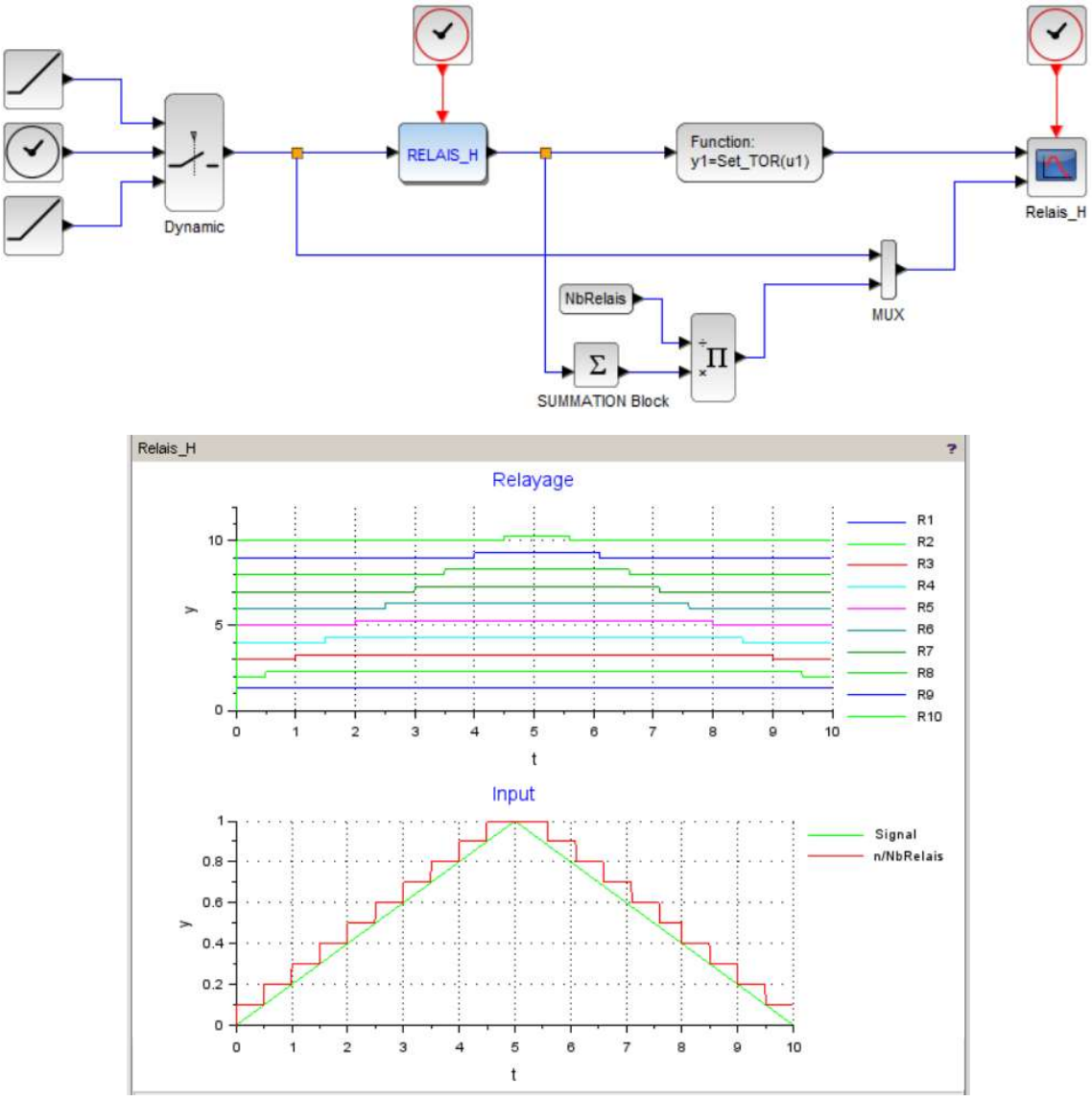
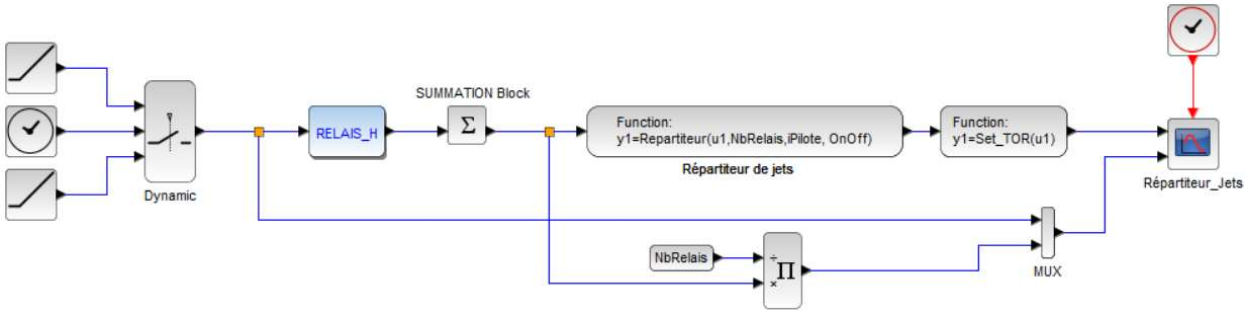


Figure 58 : Relais_H.zcos test program

Notes The **Relais_H** component is particularly well suited to the management of jet distributors for PELTON machines, allowing an efficient reduction of representation graphs



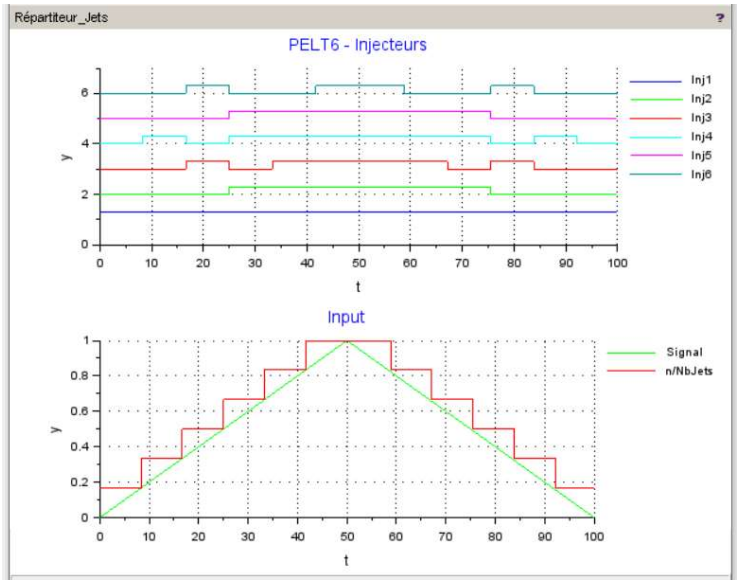
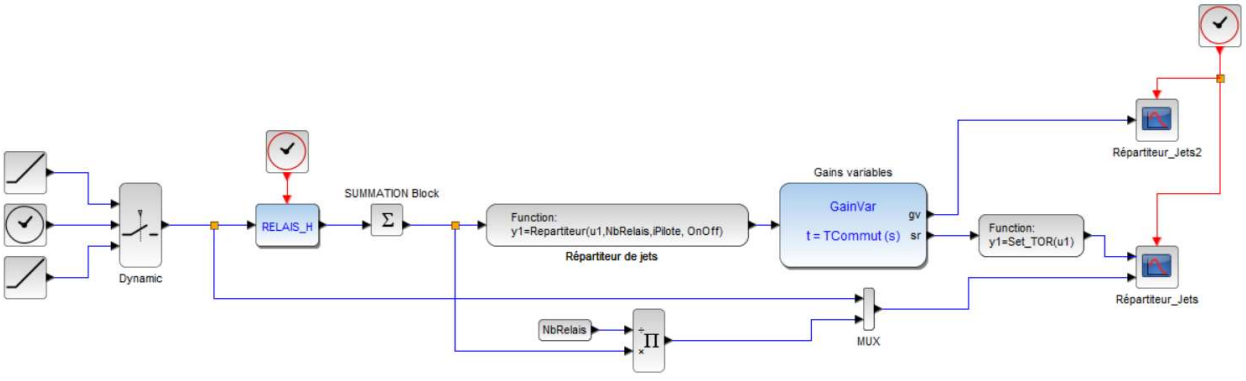


Figure 59 : Répartiteur_Jets1.zcos Jets Dispatcher test program



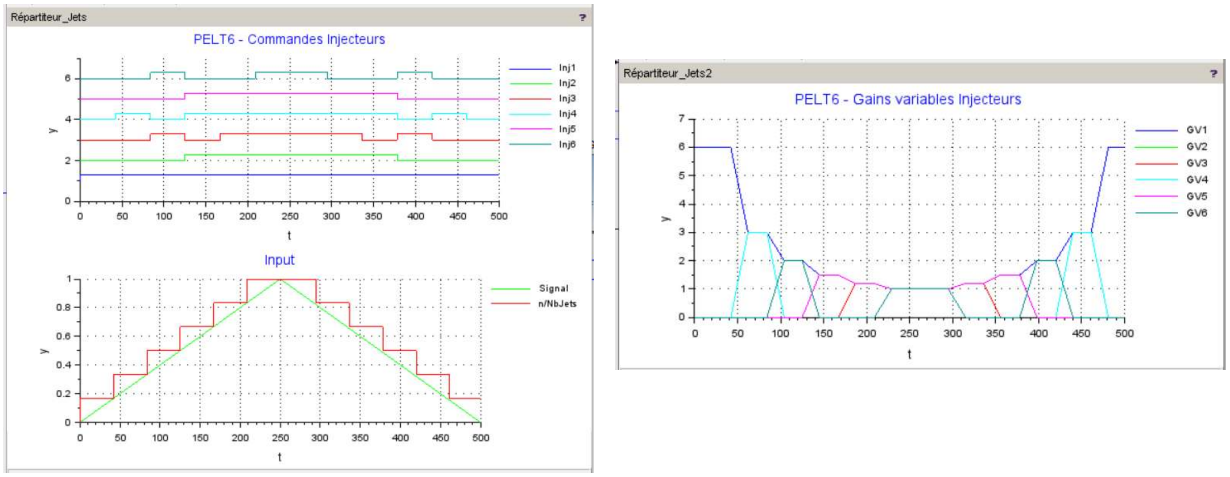


Figure 60 : Répartiteur_Jets2.zcos Test program

See also

Hysteresis_H

RELATIONOP

Multi-way relational operations

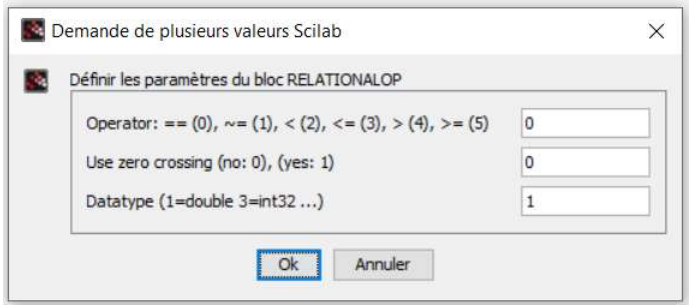
Block Screenshot



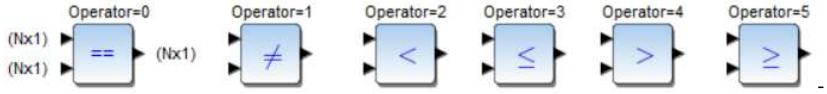
Description

The **RELATIONOP** operator performs a logical comparison on its two vectorized inputs

Parameters



Alternative forms



Interfacing function

SegPal\macros\RELATIONALOP_H.sci

Computational function

- SCI/modules/scicos_blocks/src/c/relational_op.c
- SCI/modules/scicos_blocks/src/c/relational_op_i32.c
- SCI/modules/scicos_blocks/src/c/relational_op_i16.c
- SCI/modules/scicos_blocks/src/c/relational_op_i8.c
- SCI/modules/scicos_blocks/src/c/relational_op_ui32.c
- SCI/modules/scicos_blocks/src/c/relational_op_ui16.c
- SCI/modules/scicos_blocks/src/c/relational_op_ui8.c

Test program

SegPal\demos\RELATION_OP multivariables.zcos

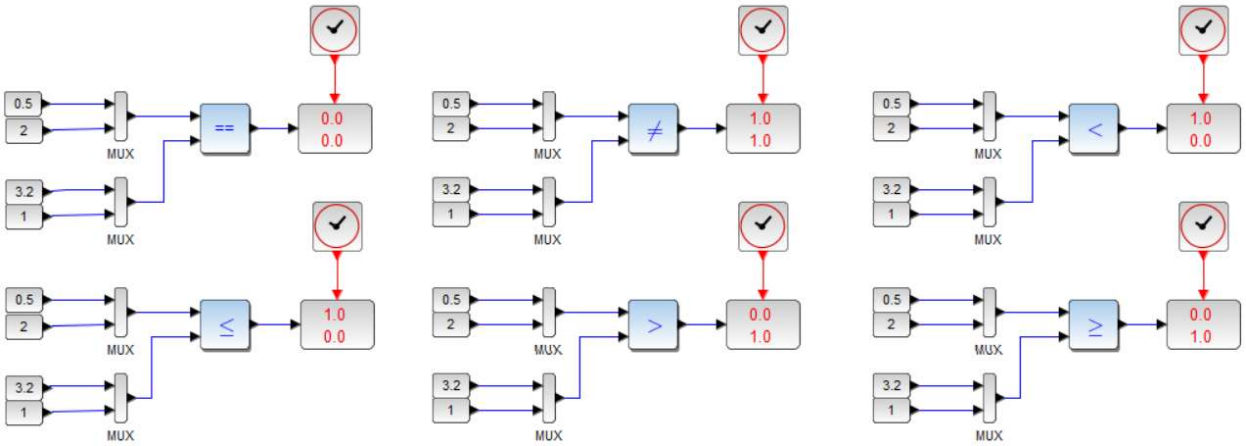
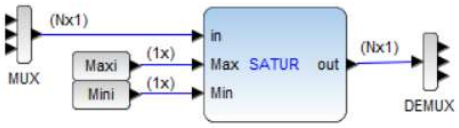


Figure 61 : RELATION_OP multivariables.zcos test program

Satur

Multivariable external saturations

Block Screenshot



Description The **Satur** block limits input signals between specified external Max and Min scalars.

Parameters (…No interface)

Alternative forms -

Interfacing function SegPal\macros\Satur.sci

Computational function SegPal\scr\c\SaturC.c

Test program SegPal\demos\Satur.zcos, Satur n lignes.zcos

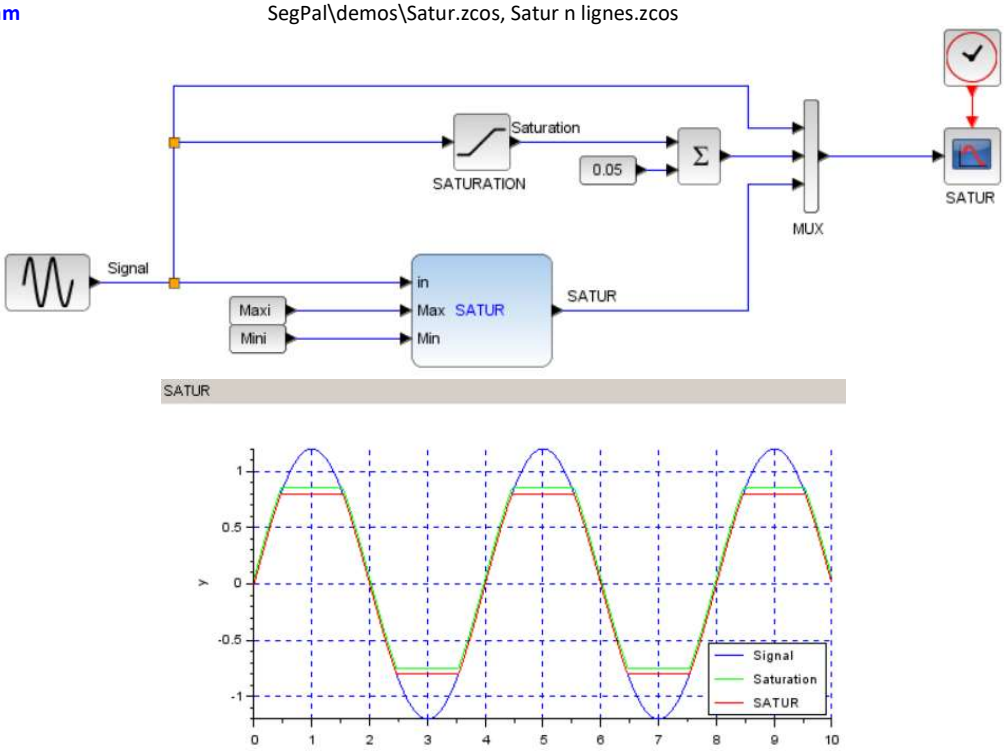


Figure 62 : Satur.zcos test program

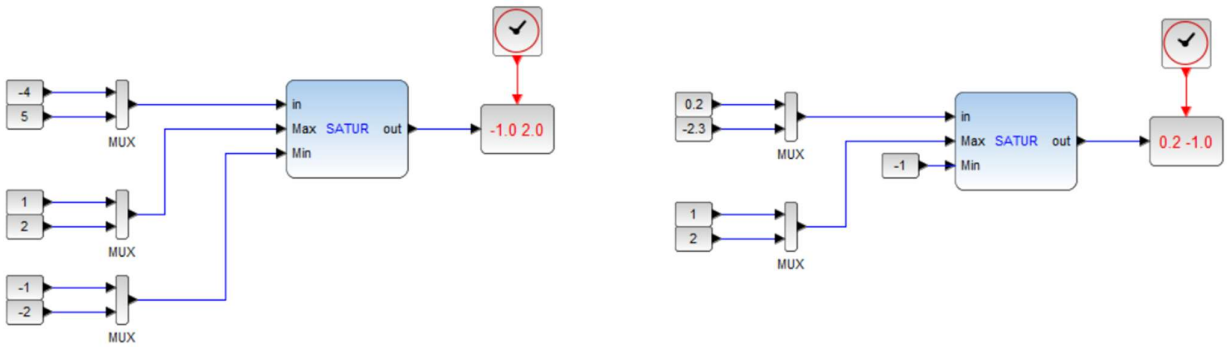


Figure 63 : Satur n lines.zcos test program

Notes

The Satur (HYDROTUR) and SATURATION (Xcos-Scilab) blocks work identically.

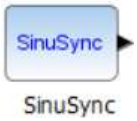
On the other hand, the advantage of the Satur block is to offer a limitation external to the definition box.

In this case, the saturation limits can be dynamic

SinuSync

Sine function with synchronization at the start time

Block Screenshot



SinuSync

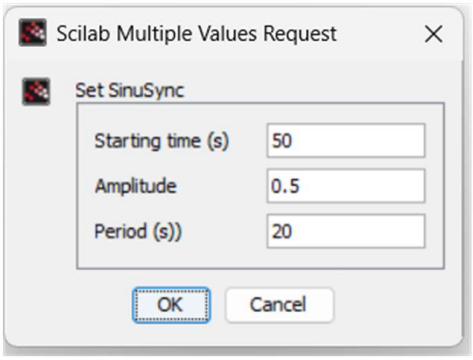
Description

The SinusSync block generates a sinusoid programmable by its period T and its amplitude A. The signal start time t_0 is also programmable.

$$y(t) = 0 \text{ if } t<t_0$$

$$y(t) = A\sin\frac{2*\pi}{T}(t-t_0) \text{ if } t \geq t_0$$

Parameters



Alternative forms

-

Interfacing function

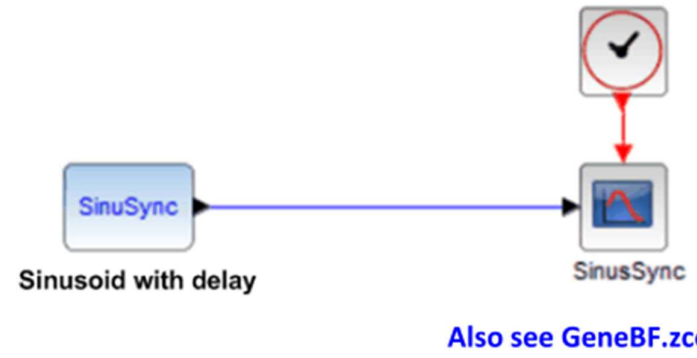
SegPal\macros\SinuSync.sci

Computational function

SegPal\scr\c\SinuSyncC.c

Test program

SegPal\demos\SinuSync.zcos, Sinus amorti.zcos



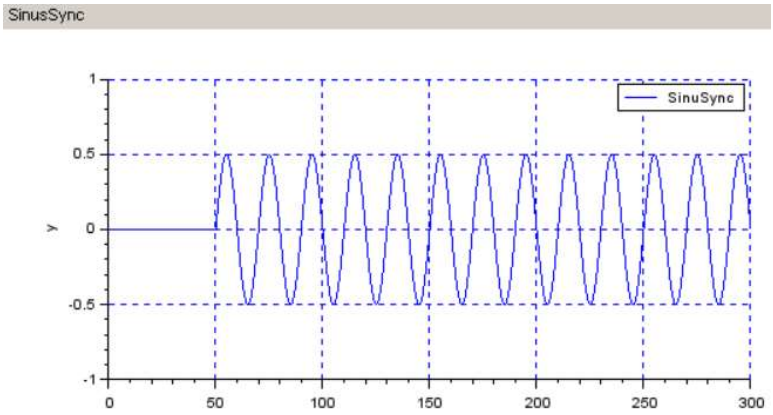


Figure 64 : SinuSync.zcos test program

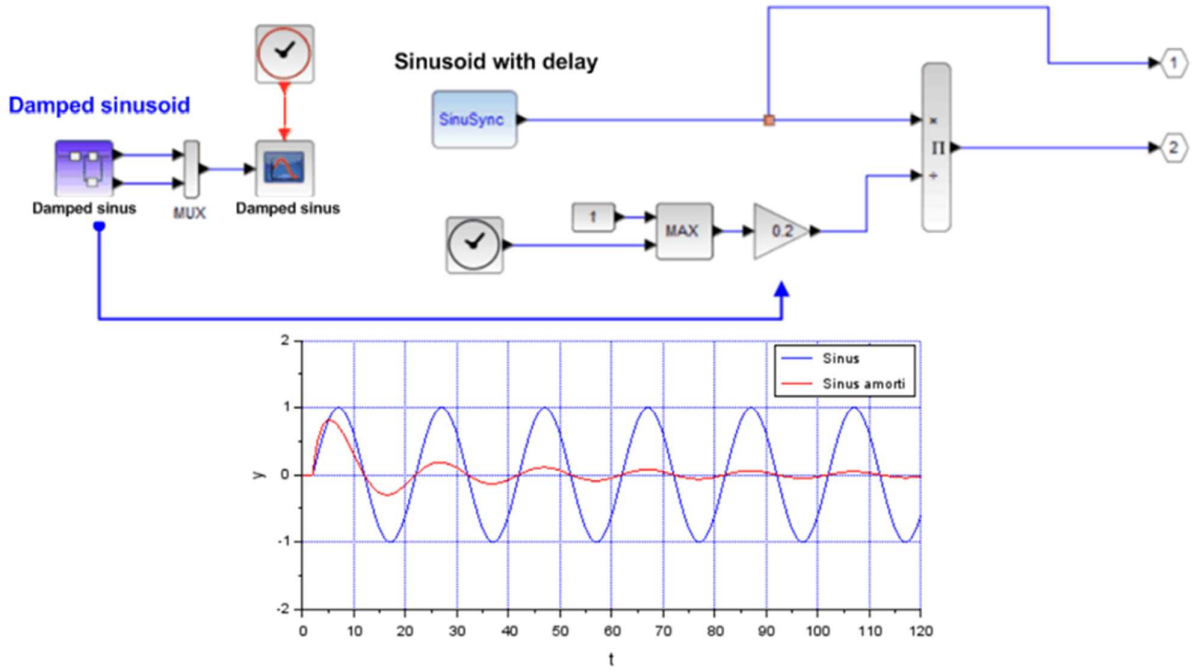


Figure 65 : Sinus amorti.zcos test program

See also : **GeneBF**

Step2

Double programmable duration step

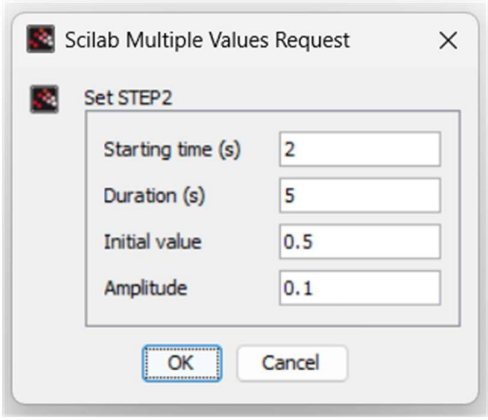
Block Screenshot



Description

The Step2 block generates two programmable steps from its interface.

Parameters



Alternative forms

-

Interfacing function

SegPal\macros\Step2.sci

Computational function

SegPal\scr\c\Step2C.c

Test program

SegPal\demos\Step2.zcos

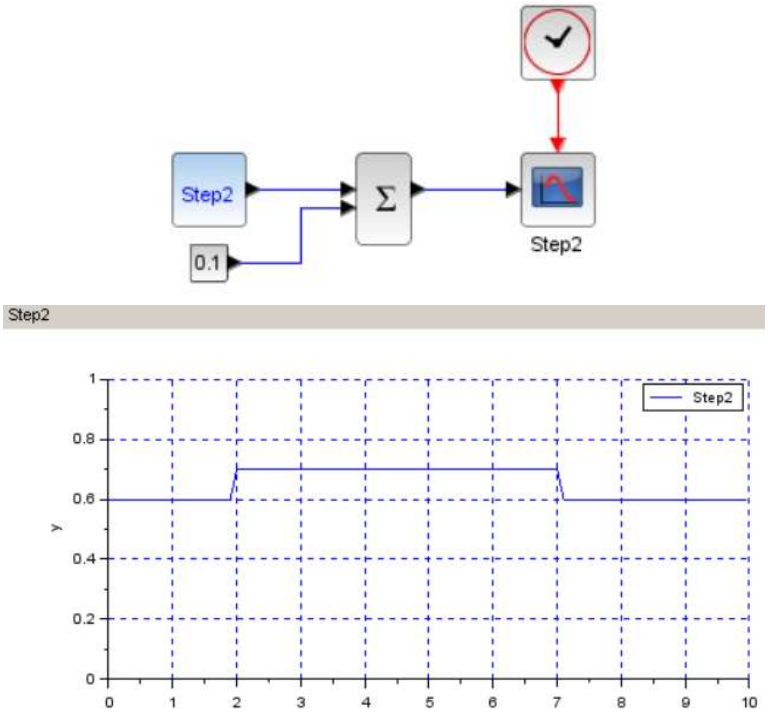


Figure 66 : Step2.zcos test program

Notes

Step 2 avoids the use of 2 Xcos-Scilab components STEP_FUNCTION  **+ 1 adder.**

TempoTrig

Retriggerable timer

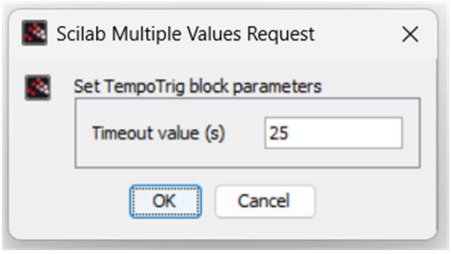
Block Screenshot



Description

The TempoTrig block generates an adjustable time delay, depending on the value of its input or its event activation.
When this timer is armed, the output decreases from the value 1 to the value 0, within the programmed time, if the in input is zero.

Parameters



Alternative forms

-

Interfacing function

SegPal\macros\TempoTrig.sci

Computational function

SegPal\scr\c\TempoTrigC.c

Test program

SegPal\demos\TempoTrig.zcos

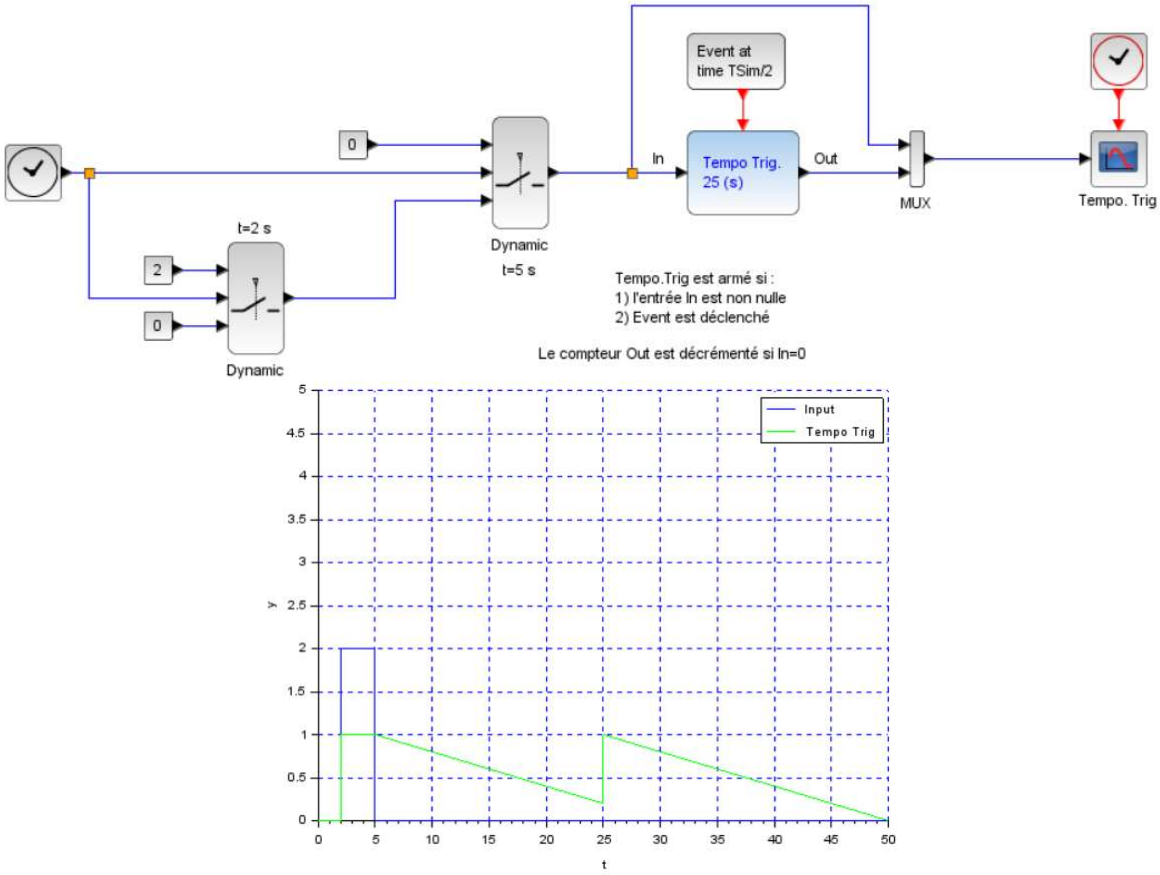


Figure 67 : Test program TempoTrig.zcos

Notes

See also EdgeTrigger, Busy.zcos test program

In some difficult cases (**HydroSim** : KAPLAN turbines Load steps at 75% of nominal power, 10% load steps), it may be necessary to reset the speed integrator by monitoring the evolution of the speed curve using a specialized SYNC1 resynchronization circuit.

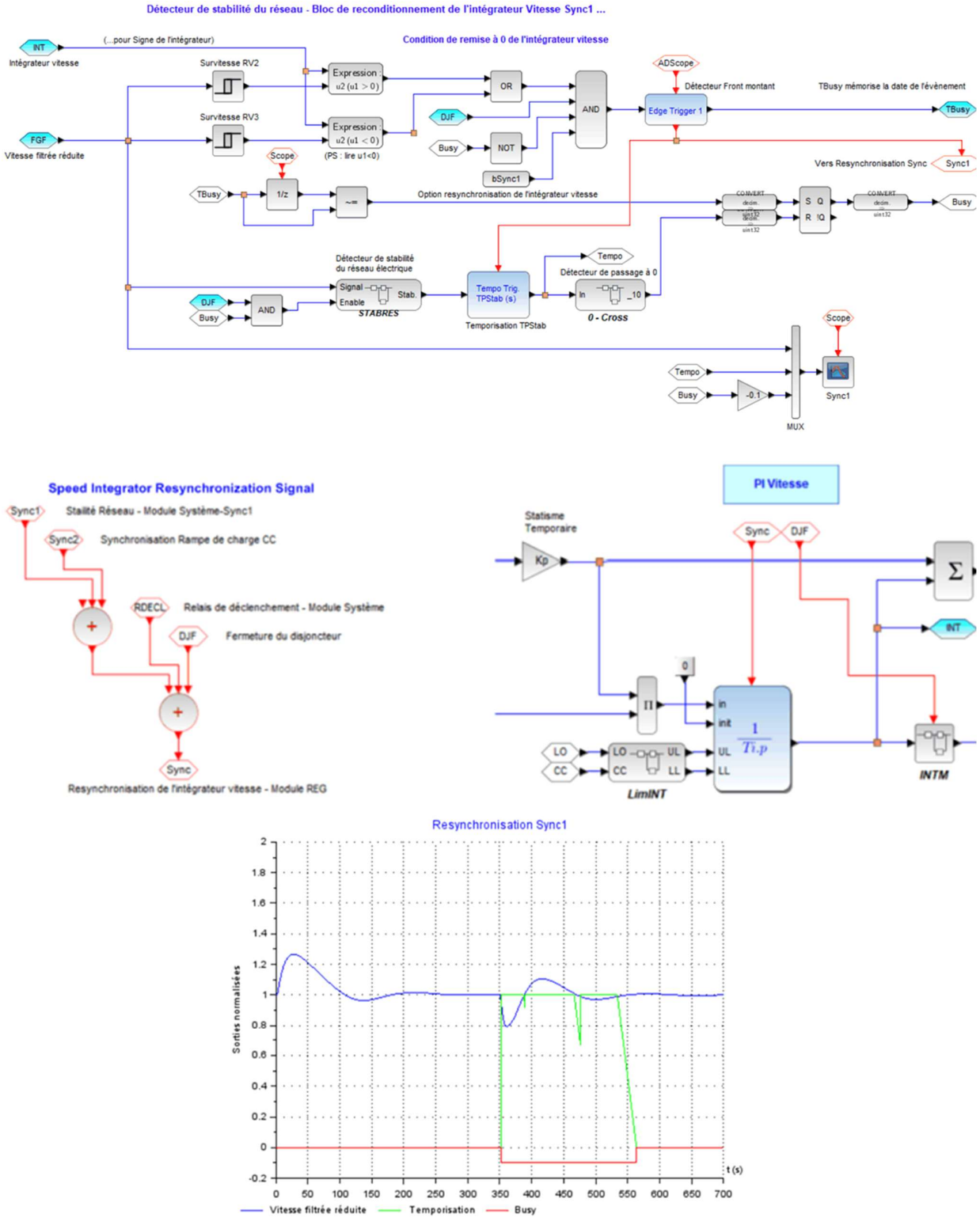
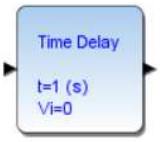


Figure 68 : Speed integrator resynchronization programs – See Sync1 Resynchronization

Time Delay

Input time offset

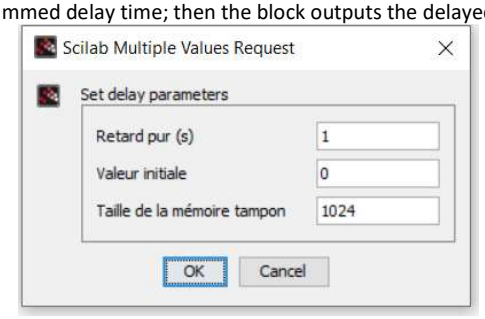
Block Screenshot



Description

The Time Delay block delays the input for a time specified in its interface. At the beginning of the simulation, the block outputs the initial input parameter until the simulation time exceeds the programmed delay time; then the block outputs the delayed input.

Parameters



Alternative forms

-

Interfacing function

SegPal\macros\Time_Delay.sci

Computational function

SCI/modules/scicos_blocks/src/c/time_delay.c (Type 4)

Test program

SegPal\demos\TimeDelay.zcos

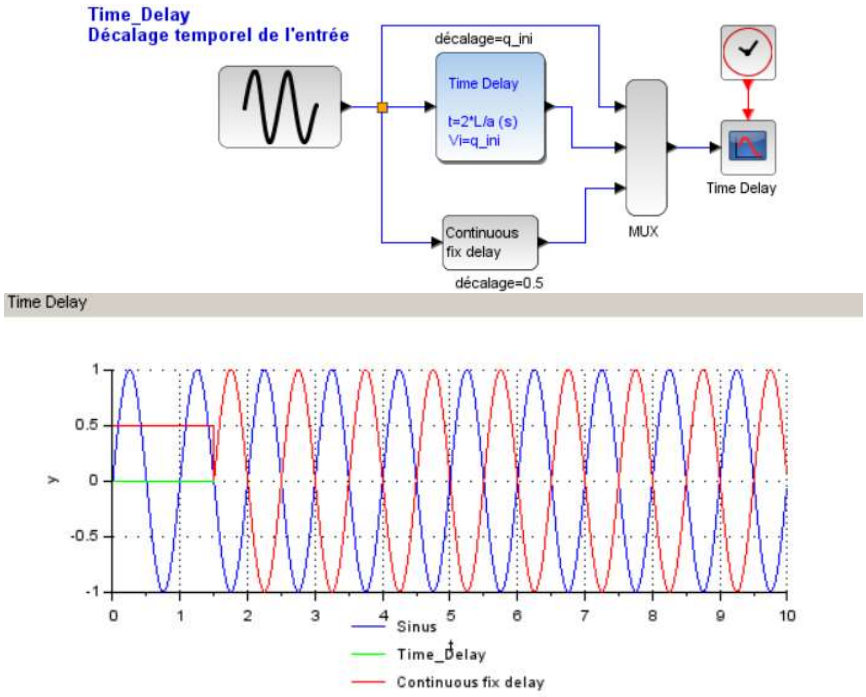


Figure 69 : TimeDelay.zcos test program

Notes

The delay time must be positive.



The Time Delay block is used in the TURBADDUC block of the FR2 and PELTON models, to initialize the initial flow rate when launching the simulation at t=0 s, and especially to generate the water hammer in the hydraulic pipes. This device allows the simulation to be launched smoothly, in steady state.

This component is like the TIME_DELAY (Continuous fix delay) component already present in the XCOS palette. The interest here is to display in the box, the programmed delay and the initialization value.

The buffer represents a space in which the time values of the input signal are stored. If the horizon is too short, error messages are displayed in the Scilab console.

It is then advisable to increase the size of the buffer to avoid any malfunction.

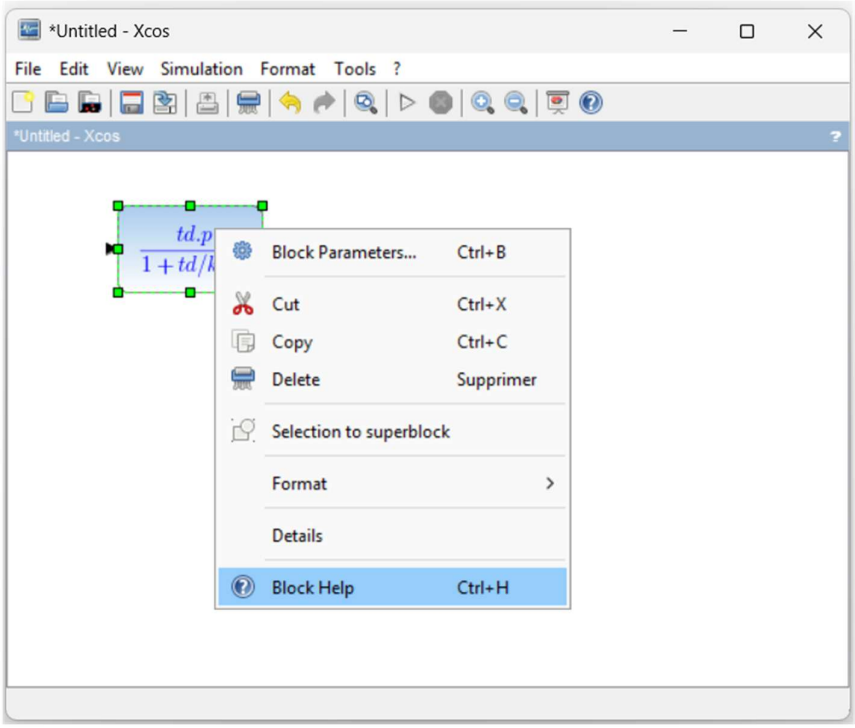
```
Consider increasing the length of buffer in delay block
delayed time=8.350000 but last stored time=9.830000
Consider increasing the length of buffer in delay block
delayed time=8.360000 but last stored time=9.840000
```

Figure 70 : Error reporting if Time Delay buffer size is not suitable

6 ONLINE HELP

6.1 Get online help for each component

Each component of the SegPal library is associated with online help, accessible by right-clicking on that component and then selecting "Online Help" from the drop-down menu displayed.



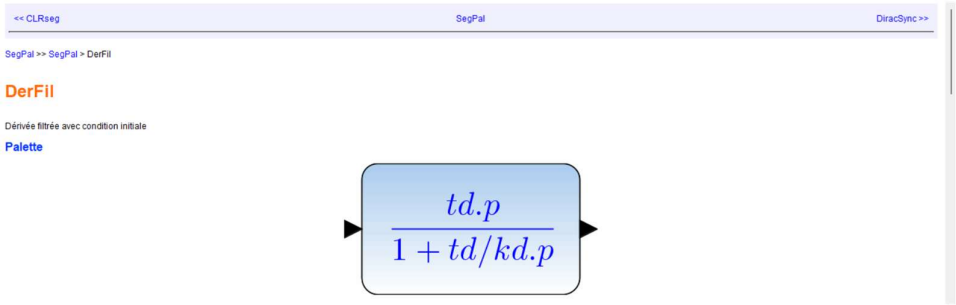


Figure 71 : Access to online help for SegPal components

6.2 Language change

The language of the online help can be changed dynamically by entering the following commands in the Scilab console:

- French language: **setdefaultlanguage("fr")**
- English language: **setdefaultlanguage("en")**

Restart Scilab after a language change.

APPENDICES

A. LISTING TYPE AFTER BUILDER'S EXECUTION.

```
--> exec('C:\SegPal\builder.sce', -1)

Construction du fichier Toolbox_names.sce dans le répertoire \etc\...

Création des macros...
-- Création de [SegPallib] (Macros) --
genlib : Traitement du fichier : CLRseg.sci
genlib : Traitement du fichier : CheckExprs.sci
genlib : Traitement du fichier : DerFil.sci
genlib : Traitement du fichier : DiracSync.sci
genlib : Traitement du fichier : EdgeTrigger.sci
genlib : Traitement du fichier : GainVar.sci
genlib : Traitement du fichier : GeneBF.sci
genlib : Traitement du fichier : Hysteresis_H.sci
genlib : Traitement du fichier : IntLim.sci
genlib : Traitement du fichier : IntSPE.sci
genlib : Traitement du fichier : Interpln_H.sci
genlib : Traitement du fichier : Memo.sci
genlib : Traitement du fichier : PSB.sci
genlib : Traitement du fichier : Pow.sci
genlib : Traitement du fichier : RELATIONALOP_H.sci
genlib : Traitement du fichier : Rampe.sci
genlib : Traitement du fichier : Relais_H.sci
genlib : Traitement du fichier : Repartiteur.sci
genlib : Traitement du fichier : Satur.sci
genlib : Traitement du fichier : SegPalError.sci
genlib : Traitement du fichier : Set_TOR.sci
genlib : Traitement du fichier : SinuSync.sci
genlib : Traitement du fichier : Step2.sci
genlib : Traitement du fichier : TempoTrig.sci
genlib : Traitement du fichier : Time_Delay.sci
Création des blocs...
Création des sources...
 "...Compilateur C"
  Génère un fichier loader
  Génère un Makefile
  Exécute le Makefile
  Compilation de PSBC.obj
  Compilation de ExponentC.obj
  Compilation de IntLimC.obj
  Compilation de RampeC.obj
  Compilation de MemoC.obj
  Compilation de SaturC.obj
  Compilation de GainVarC.obj
  Compilation de Step2C.obj
  Compilation de EdgeTriggerC.obj
  Compilation de TempoTrigC.obj
  Compilation de SinuSyncC.obj
  Compilation de DiracSyncC.obj
  Compilation de Hysteresis_HC.obj
  Compilation de Relais_HC.obj
  Compilation de Interpln_HC.obj
  Compilation de Common.obj
  Compilation de IntSPEC.obj
  Compilation de GeneBFC.obj
  Construit une bibliothèque partagée (Veuillez patienter)
  Génère un fichier cleaner
Génération de cleaner_src.sce...
Création des fichiers d'aide...
Total files without example: 21
Total generated html files: 23
Total files without example: 21
Total generated html files: 23
Total files without example: 21
Total generated html files: 23
Génération de loader.sce...
```

```
Génération de unloader.sce...
Génération de cleaner.sce...
```

Figure 72 : Listing after execution of the SegPal builder.sce constructor

LIST OF FIGURES

Figure 1 : Technical support for the HYDROTUR - SEGPAL project 4
Figure 2 : List of main HYDROTUR documents 5
Figure 3 : General functional diagram of a hydroelectric control system 8
Figure 4 : SegPal Library Components 9
Figure 5 : SEGPAL directory organization 9
Figure 6 : Main files in the SegPal directory 10
Figure 7 : Uninstalling SegPal 10
Figure 8 : « Modéliser et simuler avec XCOS » book 11
Figure 9 : Access to ATOMS Modules in Scilab libraries 11
Figure 10 : MinGW installer under Scilab 12
Figure 11 : Installing the gcc compiler 12
Figure 12 : Checking the installation of the GCC compiler-linker, version 10.3.0 64-bit 13
Figure 13 : Loading the SegPal library into Scilab – Using the loader.sce file 13
Figure 14 : Overview of the SegPal library's unit testing programs 14
Figure 15 : Runge-Kutta solver settings 14
Figure 16 : TSim simulation time setting 15
Figure 17 : Principle of numbering graphics windows 15
Figure 18 : Principle of legend insertion in test programs 15
Figure 19 : Programming windows for the "Scope Processing" function– IntLim Component 16
Figure 20 : Graphical window of the test program associated with the IntLim component 17
Figure 21 : Loading and running a test program 17
Figure 22 : Test program launch icon 17
Figure 23 : Accessing the SegPal palette in an XCOS graphics window 18
Figure 24 : Panoramic view of the additional HYDROTUR– SegPal components 18
Figure 25 : CLRseg.zcos test program 19
Figure 26 : CLRseg.zcos_Auto test program 20
Figure 27 : DerFil.zcos test program 21
Figure 28 : Accelerometer-tachometer in HydroSim governor module 22
Figure 29 : Context of the DerFil.zcos test program 22
Figure 30 : Reminder of the principle of the filtered derivative 22
Figure 31 : Test program DiracSync.zcos 23
Figure 32 : EdgeTrigger.zcos test program 25
Figure 33 : Busy.zcos test program (1/2) 26
Figure 34 : Test program Busy.zcos (2/2) 26
Figure 35 : GainVar.zcos test program – Monovariable Implantation 28
Figure 36 : GainVar_V.zcos test program – Multivariable implementation 29
Figure 37 : Principle of jet switching, with respect to the calculations of radial forces on the PELTON wheel 29
Figure 38 : Extract from systems copying SYSCOP_I of injectors (see PELTON diagram library) – Multivariable implementation 29
Figure 39 : Equivalence of the GeneBF generator with a SuperBlock 30
Figure 40 : GeneBF Generator Operator Interface 30
Figure 41 : GeneBF.zcos Generator test program 31
Figure 42 : Use of the GeneBF disruptor in the “Reseau” modules of HydroSim simulators 32
Figure 43 : Hysteresis_H.zcos test program 33
Figure 44 : Interpln_H.zcos test program 34
Figure 45 : Comparison of the integrators IntLim (SepPal) and Integral_m, Integral_f (Scilab-Xcos) 35
Figure 46 : IntLim.zcost test program 35
Figure 47 : Integrator IntLim in the HydroSim speed governor 36
Figure 48 : IntSPE Leak integrator 37
Figure 49 : IntSPE.zcos test program 37
Figure 50 : IntSPE Integrator Desaturation Time Control 37
Figure 51 : Memo.zcos test program 38
Figure 52 : Pow.zcos test program 39
Figure 53 : Approximation of the Flow/Opening law in TURBADDUC model FR2 (HydroSim) 39
Figure 54 : PSB.zcos test program 41

Figure 55 : Rampe.zcos test program 42
Figure 56 : Extract from the load recorder in HyDroSim models 43
Figure 57 : Principle of the Relay_H component 43
Figure 58 : Relais_H.zcos test program 44
Figure 59 : Répartiteur_Jets1.zcos Jets Dispatcher test program 45
Figure 60 : Répartiteur_Jets2.zcos Test program 46
Figure 61 : RELATION_OP multivariables.zcos test program 47
Figure 62 : Satur.zcos test program 47
Figure 63 : Satur n lines.zcos test program 48
Figure 64 : SinuSync.zcos test program 49
Figure 65 : Sinus amorti.zcos test program 49
Figure 66 : Step2.zcos test program 50
Figure 67 : Test program TempoTrig.zcos 51
Figure 68 : Speed integrator resynchronization programs – See Sync1 Resynchronization 52
Figure 69 : TimeDelay.zcos test program 53
Figure 70 : Error reporting if Time Delay buffer size is not suitable 54
Figure 71 : Access to online help for SegPal components 55
Figure 72 : Listing after execution of the SegPal builder.sce constructor 57

Personal notes