HYDROTUR

HYDROTUR - SEGPAL
Simulator for Hydraulic Turbines

Graphics components library for XCOS
Reference Guide - Version 2.2

by Pierre Perrichon

Scilab

CONTENTS

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.

Type de document	Références
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

Figure 1 : Technical support for the HYDROTUR - SEGPAL project

INDEX

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 schematic libraries and simulators.

SEGPAL is initially dedicated to the HYDROTUR 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.

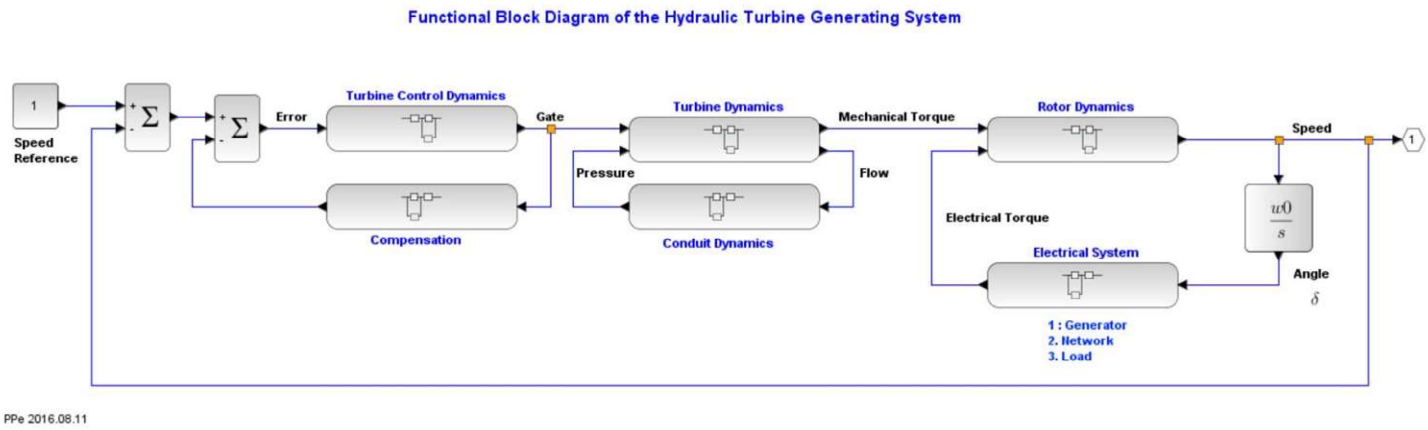


Figure 2 : Principe de base des simulateurs HYDROTUR pour turbines hydrauliques

☞ **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:

Composant	Signification	Demo
CLRseg	First order low pass filter with initialization at t=0	CLRseg.zcos
DerFil	Filtered derivative with initial condition	DerFil.zcos
DiracSunc	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 Répartiteur_Jets1.zcos Répartiteur_Jets2.zcos
RELATIONALOP_H	Multi-form relational operations	RELATION_OP 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	TimeDelay.zcos

Figure 3 : SegPal Library Components

2 SUPPLY

SegPal is provided as a zip file: SegPal.zip.

SegPal.zip can be unpacked to any directory or subdirectory chosen by the user.

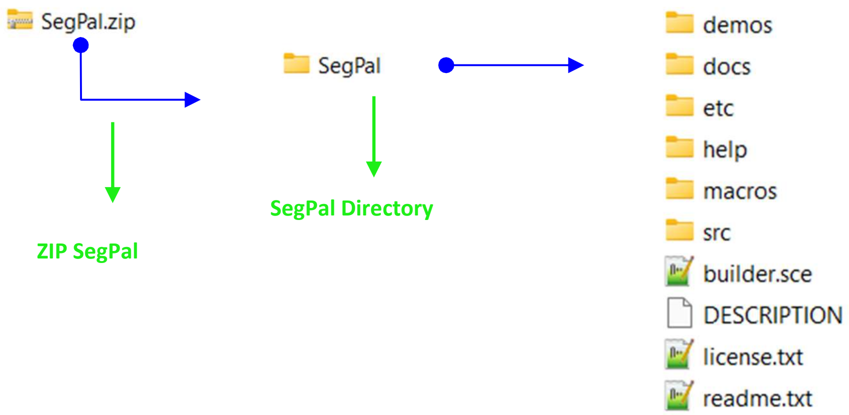


Figure 4 : SegPal Directory Organization

Directory or File	Content
demos	Set of xcos demonstration programs illustrating each SegPal component
docs	Documentation and installation guides (English/French): • HYDROTUR - SEGPAL - Guide de référence version 2.2.pdf • HYDROTUR - SEGPAL - Reference guide version 2.2.pdf
etc	Initialization (SegPal.start file) and finalization (SegPal.quit file). Scripts used in loading the SegPal library “loader.sce” or unloading it “unloader.sce”
help	• Help files in XLM format divided into subdirectories in French (fr) and English (en) • Set of images used in the online help of the components (subdirectory “gui”)
macros	Macros (.sci) Set of source files written in Scilab language – Description of the graphical interfaces of each component (form of the component + parameterization operator interface)
src/c	Set of source files written in C language. Execution codes for each component

builder.sce	SEGPAL palette Main builder
DESCRIPTION	Description of the library palette
license.txt	Declaration of conformity to the CeCILL license Version 2.1
readme.txt	Recommendation for installing SegPal on the computer

Figure 5 : Contents of the SegPal.zip file

3 BUILDING AND LOADING THE SEGPAL LIBRARY

3.1 Foreword

The building of the SEGPAL library requires a good understanding of the work [2][1].

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

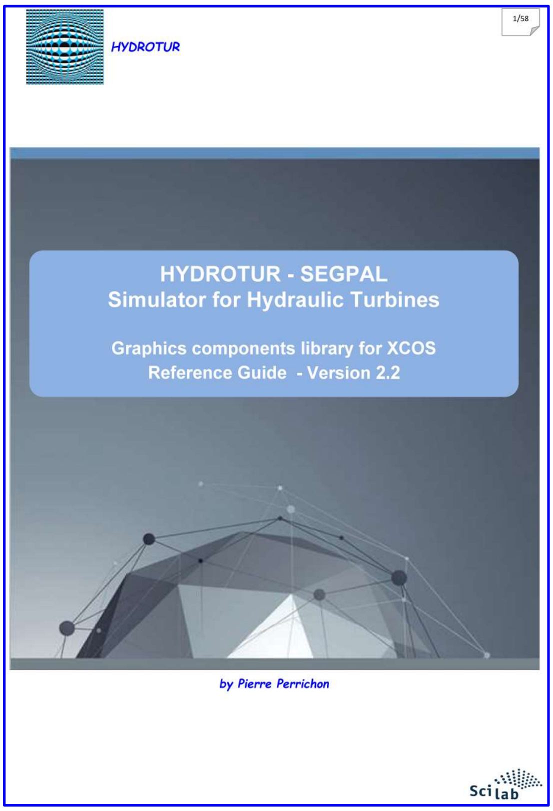


Figure 6 : Book Modeling and simulating with XCOS

3.2 Prerequisites – Installation of the C-C++ MinGW compiler in Scilab – ATOMS module

The source codes of the executives linked to the functional boxes are mainly written in C language, and available in the **SegPal\scr\c** directory.

It is therefore 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 in the list of modules:

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

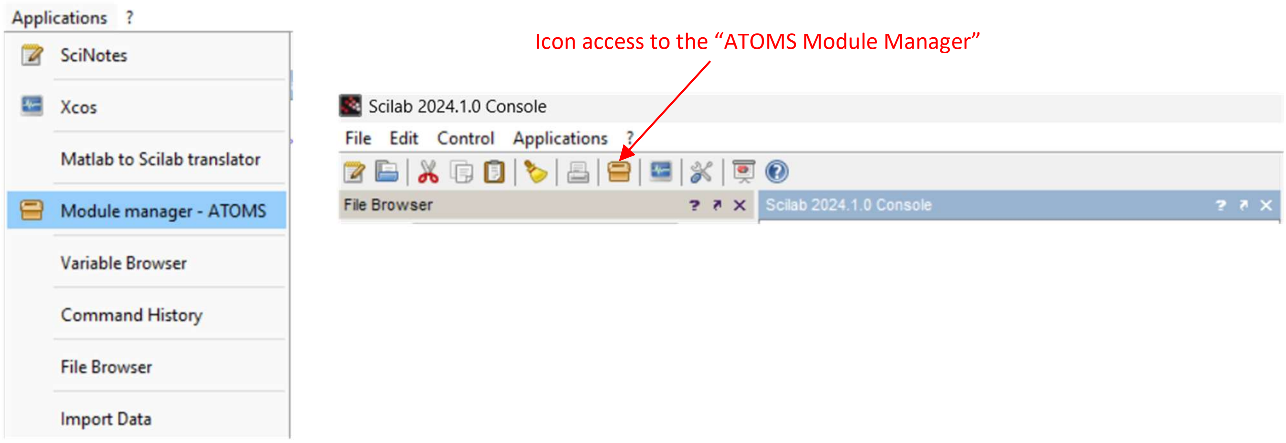


Figure 7 : Access to ATOMS Modules of Scilab Libraries

1. Click on the "All Modules" icon to get the "Main categories" display

2. From the alphabetical drop-down list, select the "MinGW" component

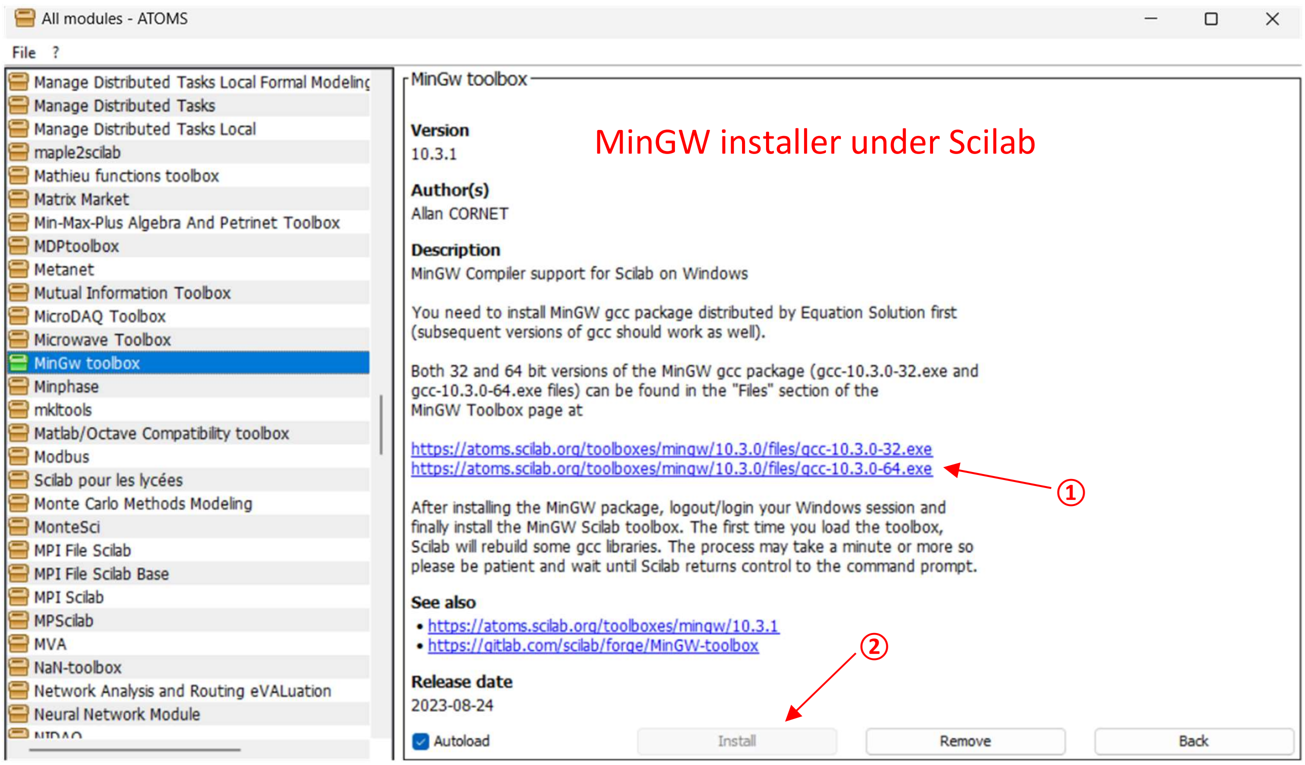


Figure 8 : MinGW installer under Scilab

3. Get the **64-bit MinGW compiler in the version indicated in** ① and install it in the directory chosen by the user:

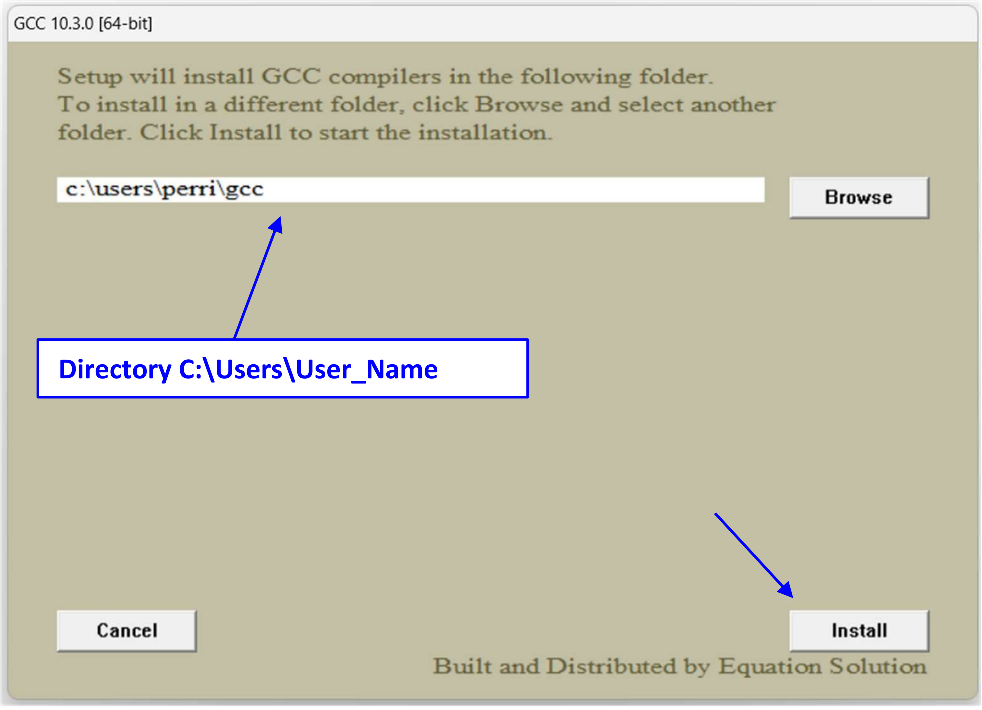


Figure 9 : Installing the gcc compiler

Typically, unless the user is changed, a default directory is created at “C:\Users\User_Name».

At the end of the installation, you can check that “gcc” has been installed correctly.:

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

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

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

Figure 10 : Verifying the installation of the GCC compiler-linker Version 10.3.0 64-bit

4. Complete the MinGW installation by clicking “Install” as shown in ② on *Figure 8*.

5. Close the Scilab session and start a new session to complete the installation

3.3 Builder builder.sce

The script **builder.sce** file, located in the SegPal directory, creates the SegPal library and provides all the files needed to use it.:

- Compilation of all .sci macros, associated with the macros subdirectory of SegPal\macros
- Compilation of all C language sources associated with the functional boxes, in the SegPal\src\c directory
- Creation of help files in French and English, providing online help on each component with a simple right-click on a component.

Additionally, the builder.sce builder provides and adds the following files to the SegPal directories:

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

A listing, obtained by screen copy after launching the buider.sce constructor in Scilab, is provided in appendix ***Erreur ! Source du renvoi introuvable.***

3.4 Loader loader.sce

3.4.1 Loading

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

Running this script loads the **SegPal** library into Scilab. If successful, the **Scilab** console will display a list of the operations performed.:

```
Start SegPal Version 2.2 : Additional components for XCOS graphic library
        Load macros
        Load palette
        Load simulations functions
        Load help
```

Figure 11 : Loading the SegPal library into Scilab – Using the loader.sce file

3.4.2 Troubleshooting

☞ **The loader.sce file is not backward compatible depending on the Scilab versions. For example, it is not possible to load SegPal created with Scilab 2024.1.0 into Scilab 2024.1.0, or previous versions (Scilab scratch)**

It is then necessary to relaunch the builder.sce constructor in the appropriate version of Scilab.

4 TEST PROGRAMS ARCHITECTURE

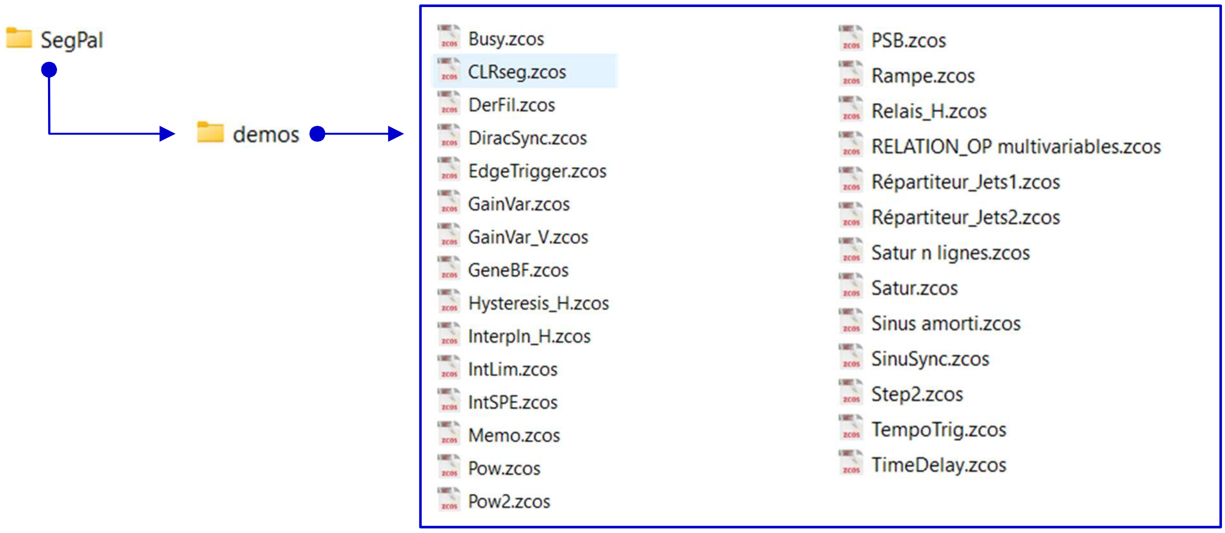


Figure 12 : Overview of the unit test programs of the SEGPAL library[2]

4.1 Launching a test program

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

This is executed directly by clicking the dedicated tool in the XCOS menu bar:

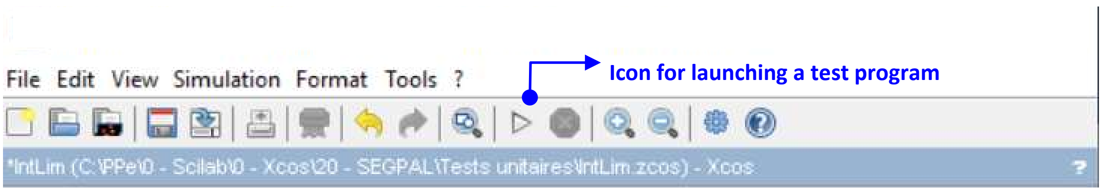


Figure 13 : Icon for launching a test program

[2] The list presented here is not exhaustive. Consult the "demos" directory for the complete list.

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

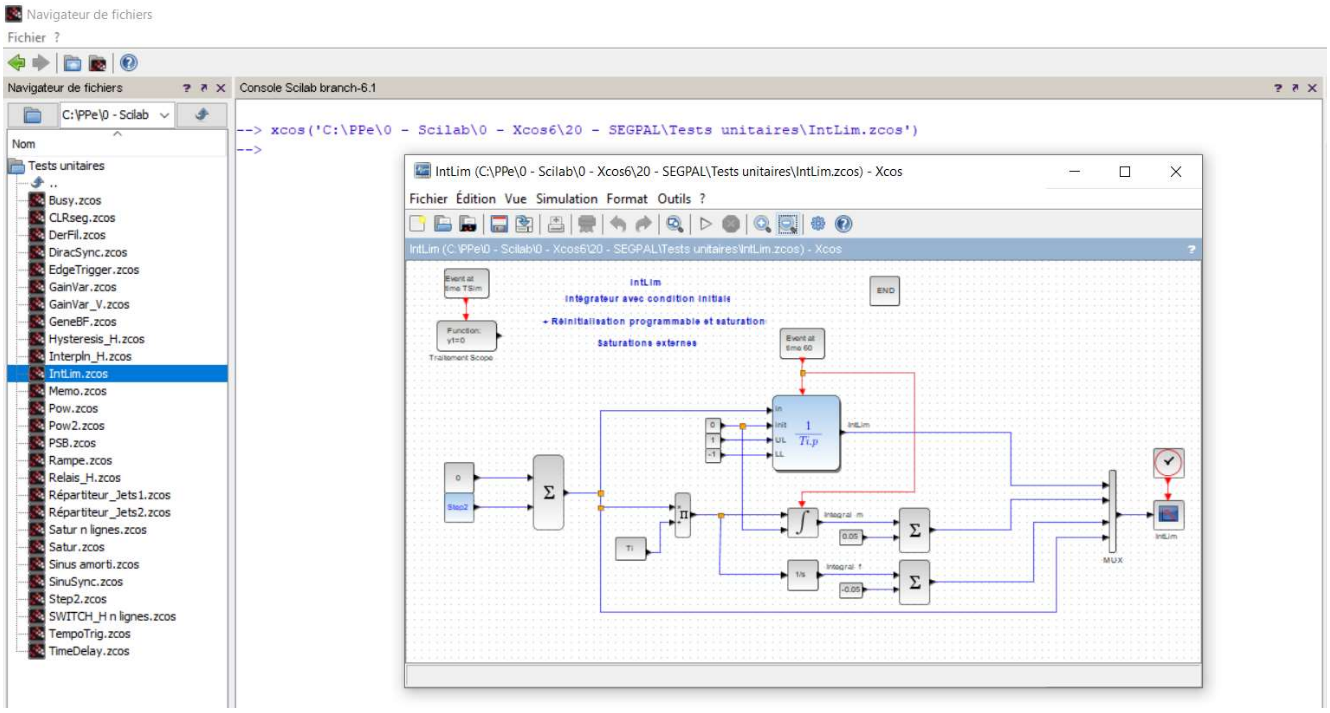


Figure 14 : Loading and Running a Test Program

Then launch the test program using the dedicated tool as shown in *Figure 13.*

5 SPECIFIC COMPONENTS

5.1 General view

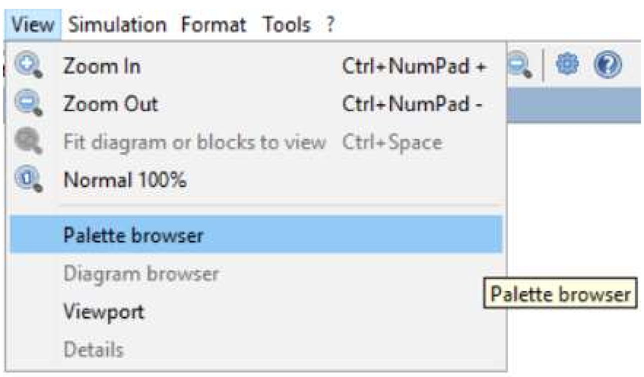


Figure 15 : Accessing the **SEGPAL** palette in an XCOS graphics window

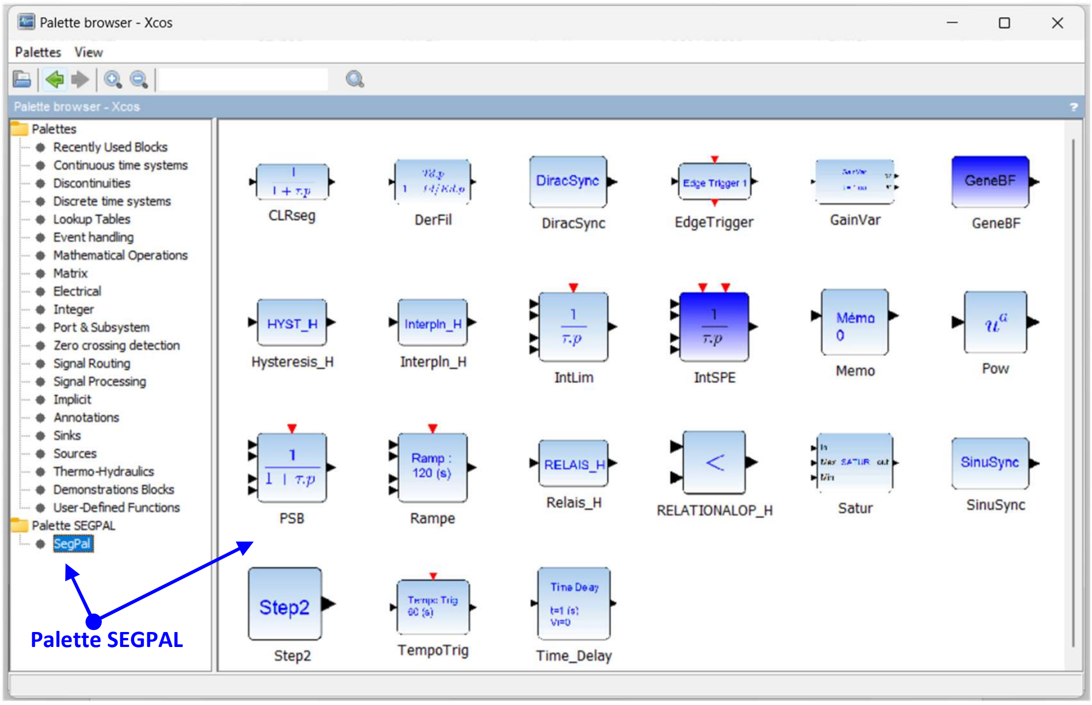
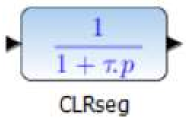


Figure 16 : Panoramic view of the additional components of the SEGPAL Block Screenshot

5.2 Description

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

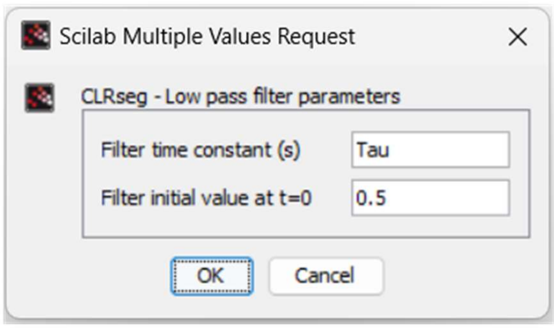
Block Screenshot



Description

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

Parameters



Alternative forms

Interfacing function SEGPAL\macros\CLRseg.sci

Computational function SCI/modules/scicos_blocks/src/c/csslti4.c (Type 4)

Test program SEGPAL\demos\CLRseg.zcos

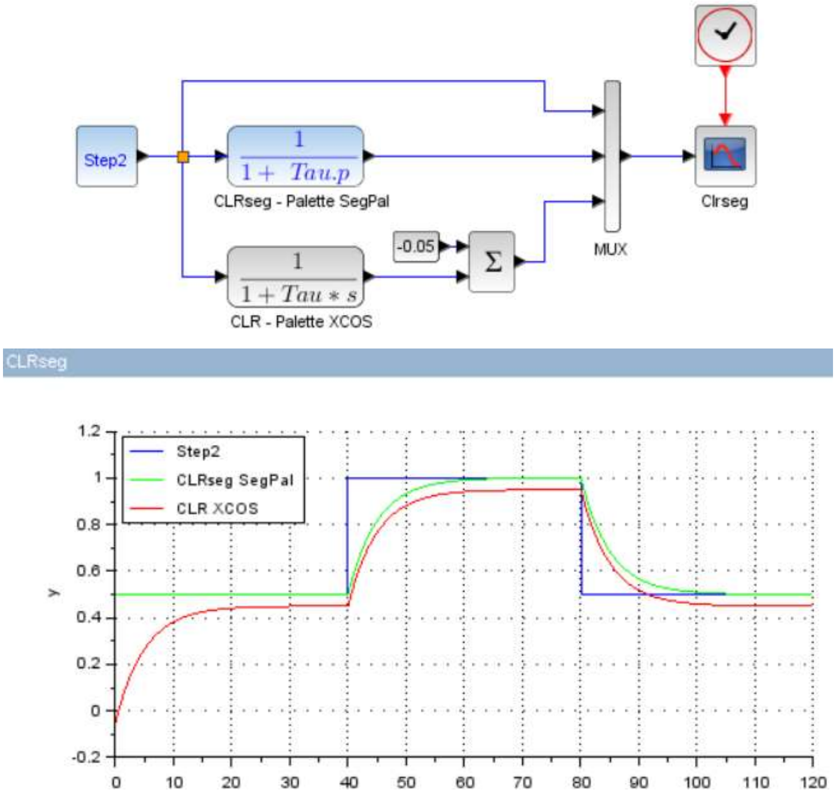
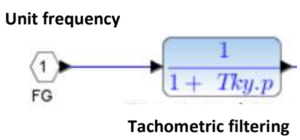


Figure 17 : CLRseg.zcos test program

Notes

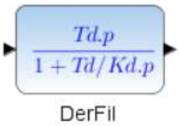


The CLRseg block is for example used in the REGxx models of the speed regulator to filter the speed of the group. Depending on the state of the circuit breaker, the initialization value is 0 if the group is stopped, or the nominal value if the simulation starts with the circuit breaker closed, group coupled to the network.
This device allows the simulation to be launched smoothly and in steady state.

DerFil

Filtered derivative with initial condition

Block Screenshot



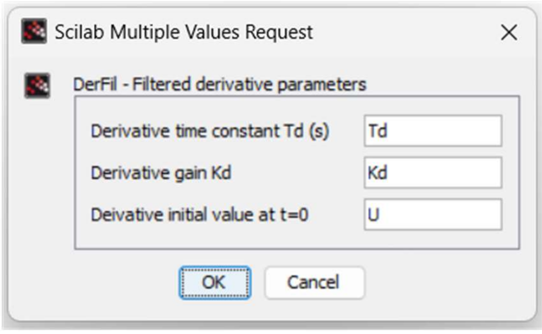
Description

The DerFil block represents a filtered derivative. The time constant Td and the gain Kd of the derivator are programmable. It is possible, if it is known, to program the initial input of the filter at t=0 in order 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 derivative action can be canceled by programming $K_d = 0$, if necessary.

Parameters

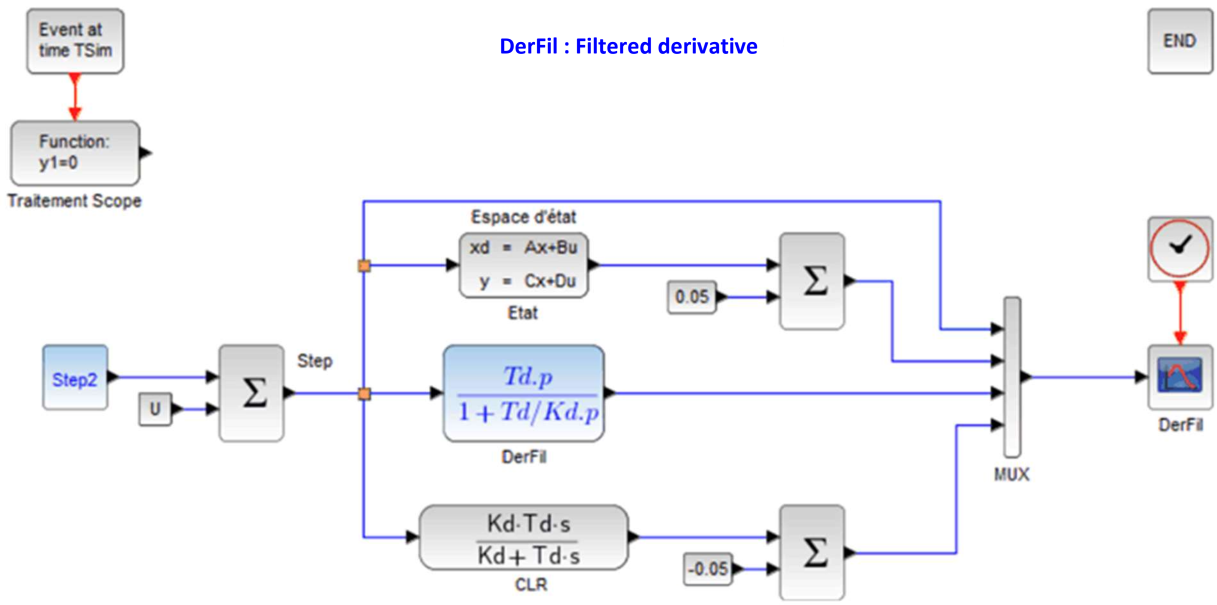


Alternative forms

Interfacing function SEGPAL\macros\DerFil.sci

Computational function SCI/modules/scicos_blocks/src/c/csslti4.c (Type 4)

Test program SEGPAL\demos\DerFil.zcos



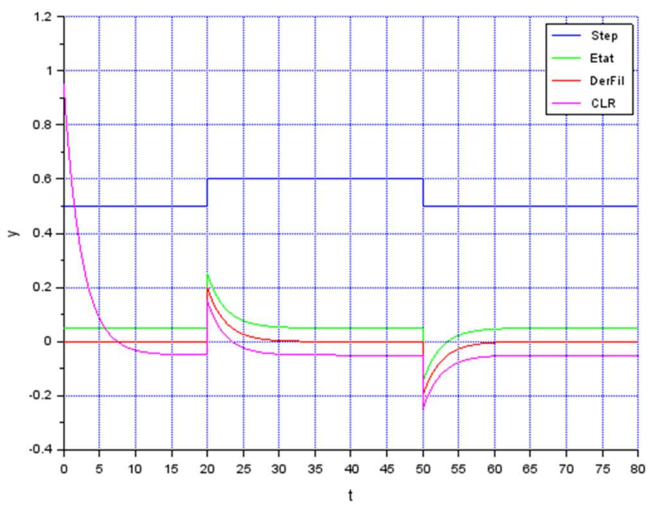
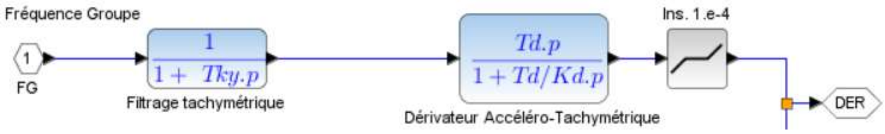


Figure 18 : DerFil.zcos test program

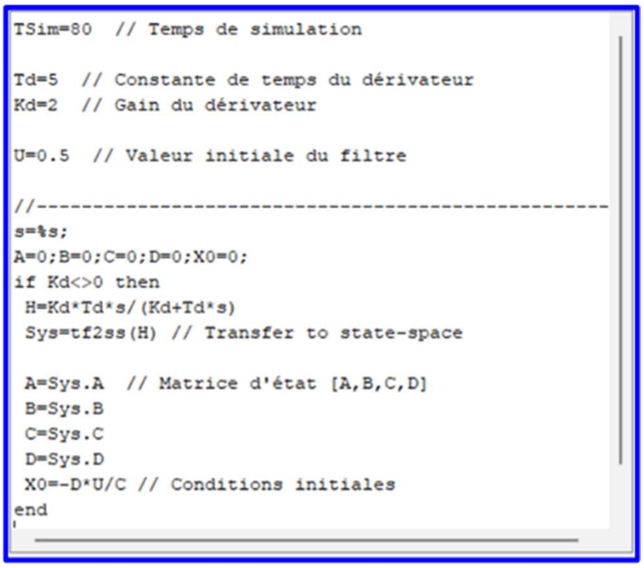
Notes



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 produced in Figure ***Figure 18*** shows three possible constructions of the same differentiator filter.

The state representation in matrix form (A, B, C, D) is calculated in the preamble when launching the simulation (***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 19 : DerFil.zcos Test program context

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

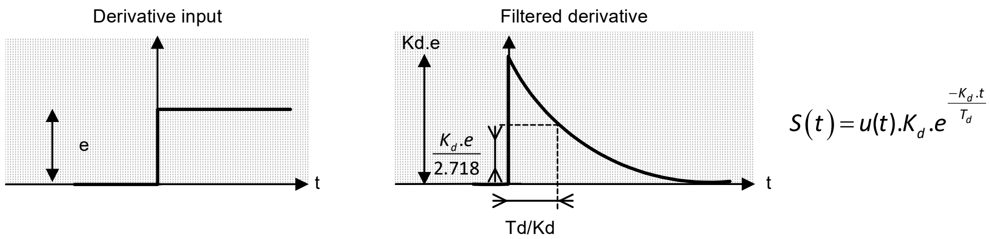


Figure 20 : Reminder of the principle of the filtered derivator

DiracSync

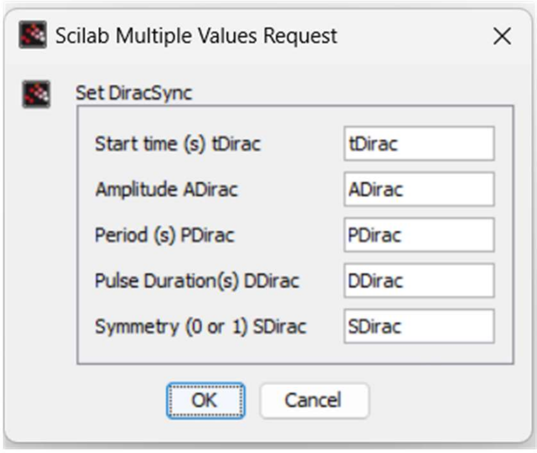
Generation of Dirac comb synchronized to a start date

Block Screenshot

Description

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

Parameters

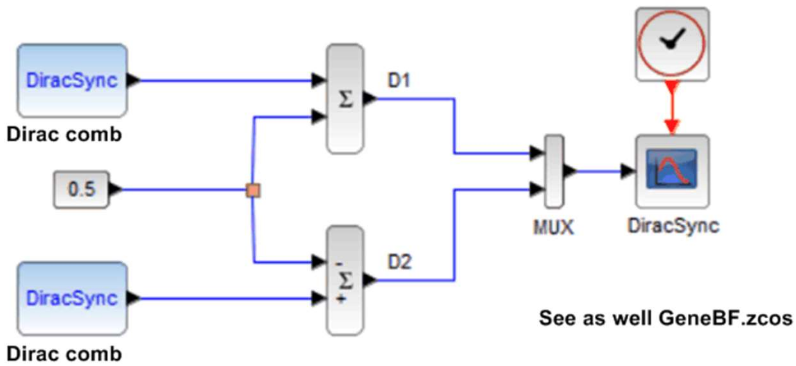


Alternative forms -

Interfacing function SEGPAL\macros\DiracSync.sci

Computational function SEGPAL\scr\c\DiracSyncC.c

Test program SEGPAL\demos\DiracSync.zcos



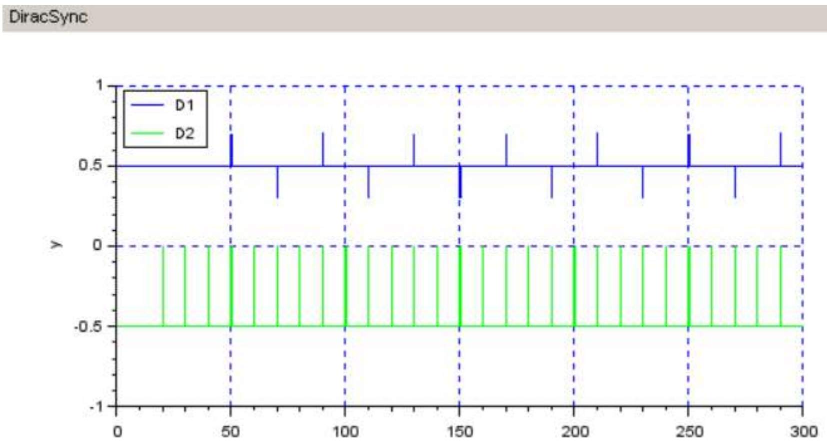
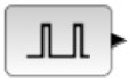


Figure 21 : DiracSync.zcos test program

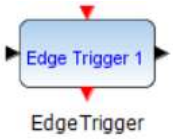
Notes

The “Symmetry” option allows you to reverse the direction of the Dirac at each sampling period (channel D1 in the graph above).

The DiracSync block is similar to the PULSE block  , standard in the XCOS Sources palette.
The interface is simplified here, and adapted to the HYDROTUR project.

EdgeTrigger — Edge detector of an input square signal

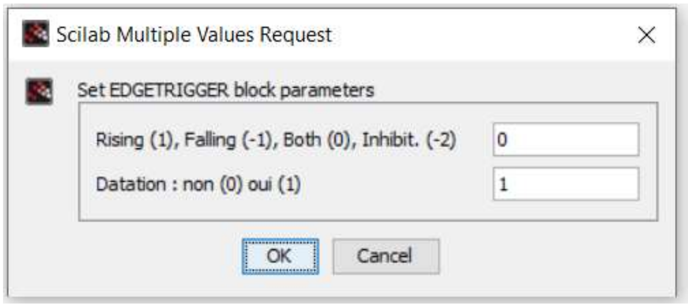
Block Screenshot



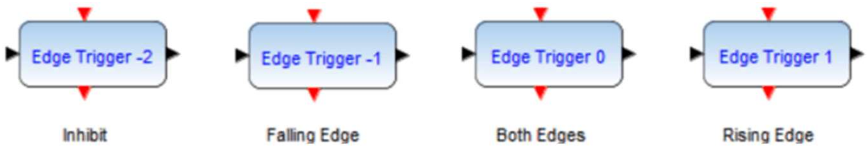
Description

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.
The selected option is entered directly into the component shape.

Parameters



Alternative forms

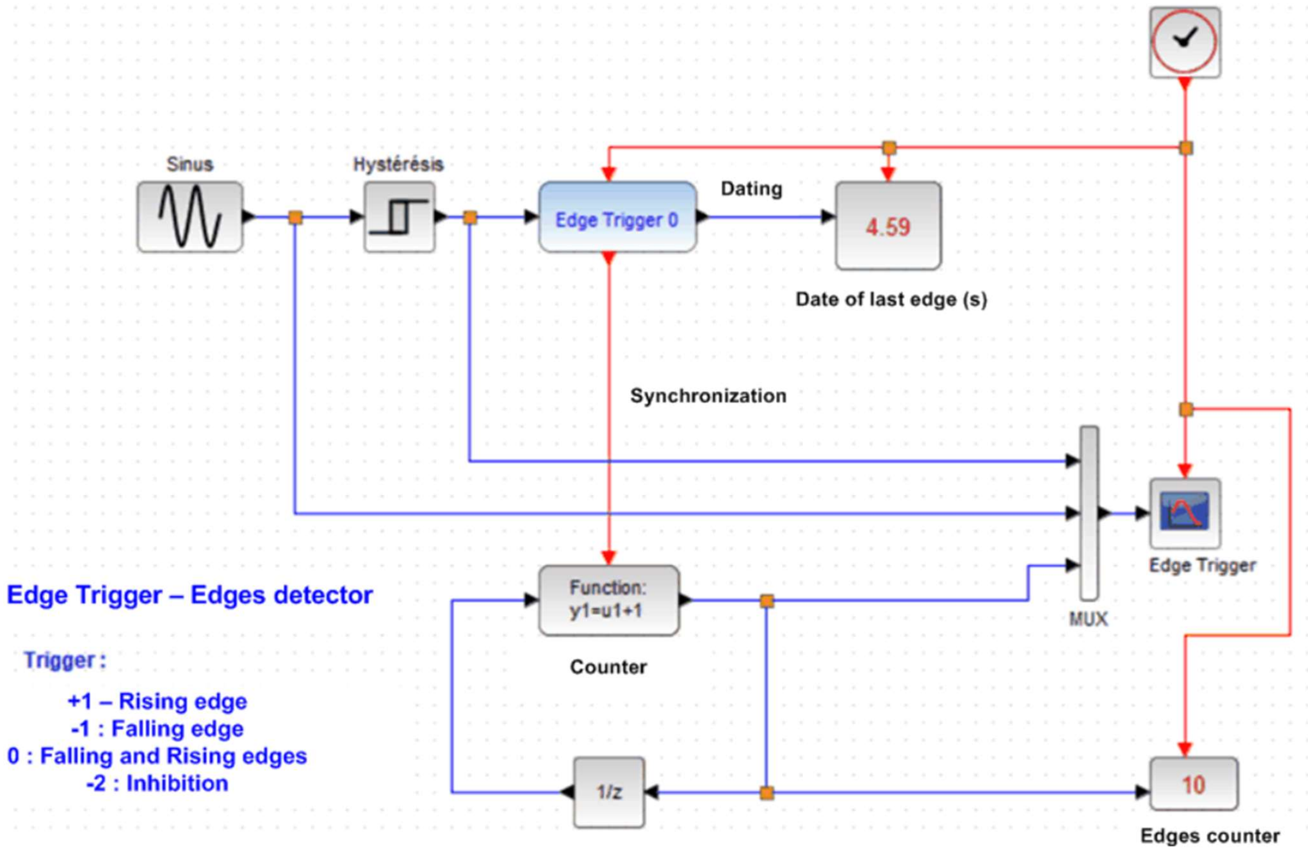


-

Interfacing function SEGPAL\macros\EdgeTrigger.sci

Computational function SEGPAL\scr\c\EdgeTriggerC.c

Test program SEGPAL\demos\EdgeTrigger.zcos



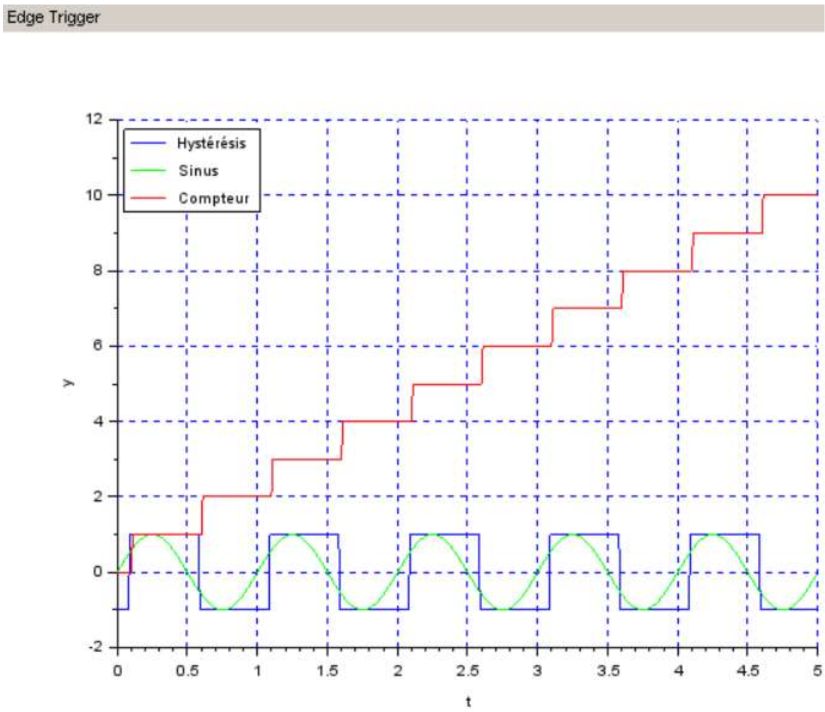


Figure 22 : EdgeTrigger.zcos test program

Notes

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

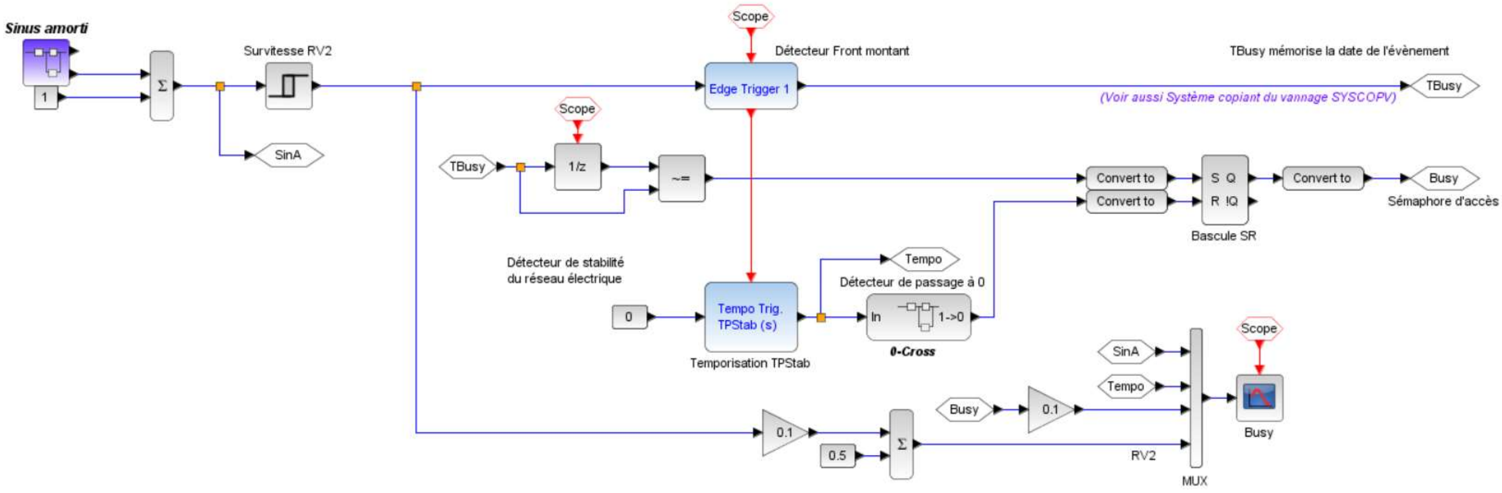


Figure 23 : Busy.zcos test program (1/2)

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.

In 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.

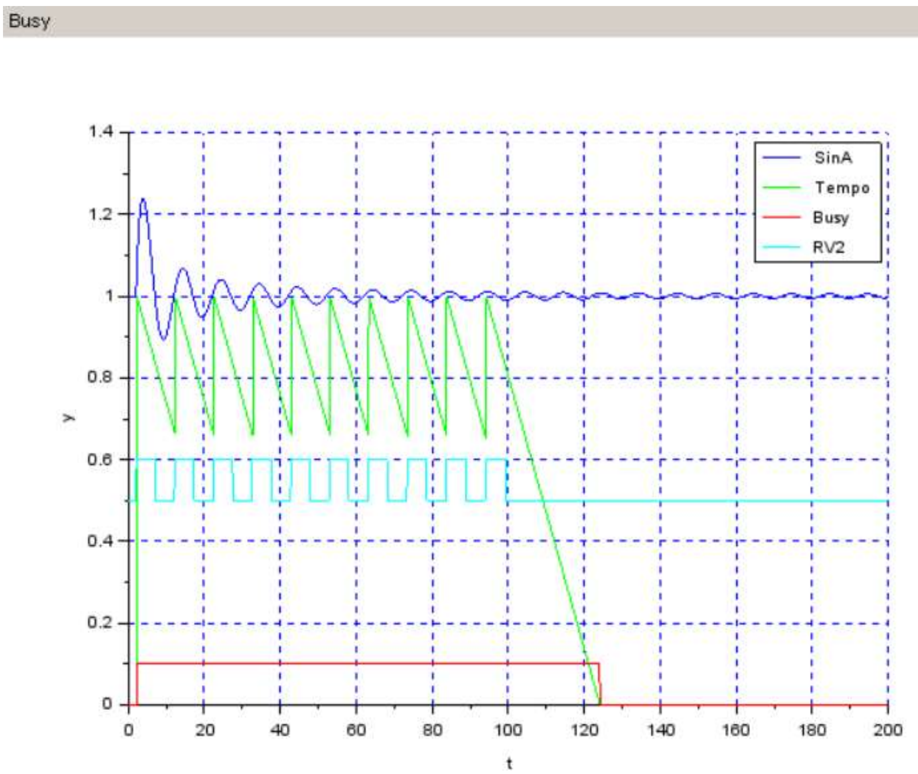


Figure 24 : Busy.zcos test program (2/2)

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.

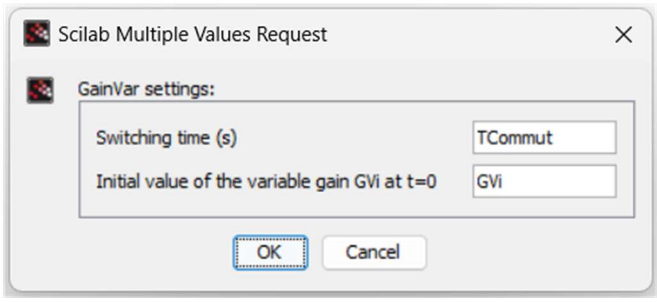
Block Screenshot

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 "**HYDROTUR - Programming guide**"

Parameters

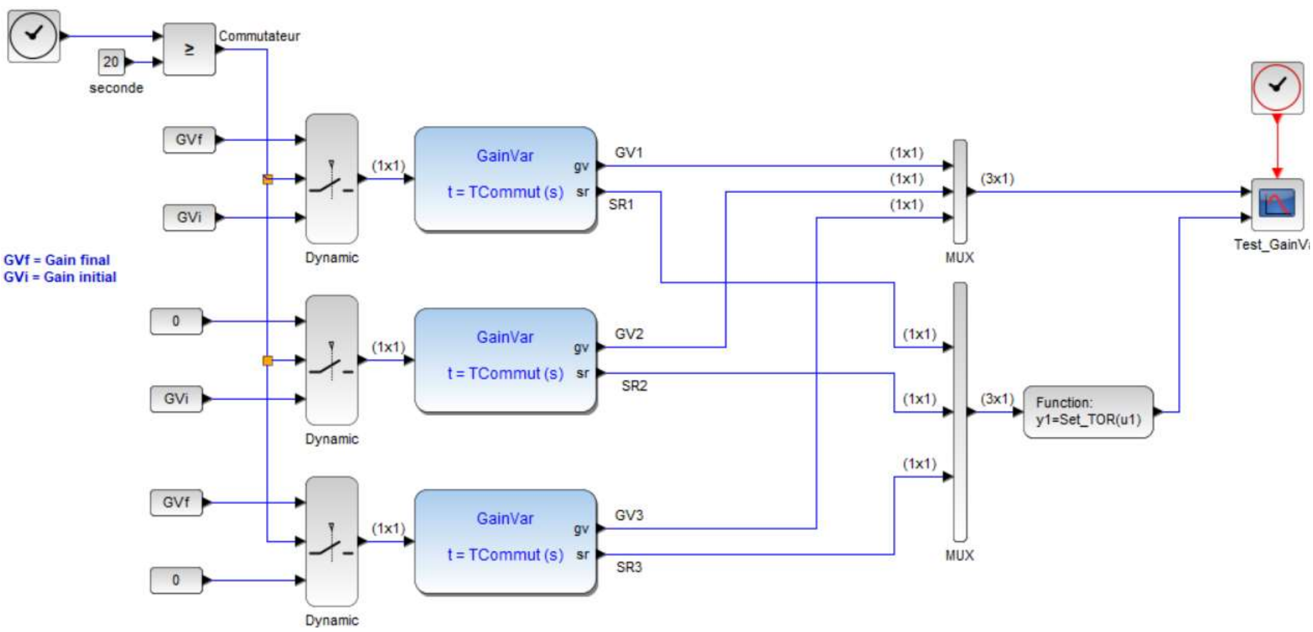


Alternative forms -

Interfacing function SEGPAL\macros\GainVar.sci

Computational function SEGPAL\scr\c\GainVarC.c

Test program SEGPAL\demos\GainVar.zcos et GainVar_V.zcos



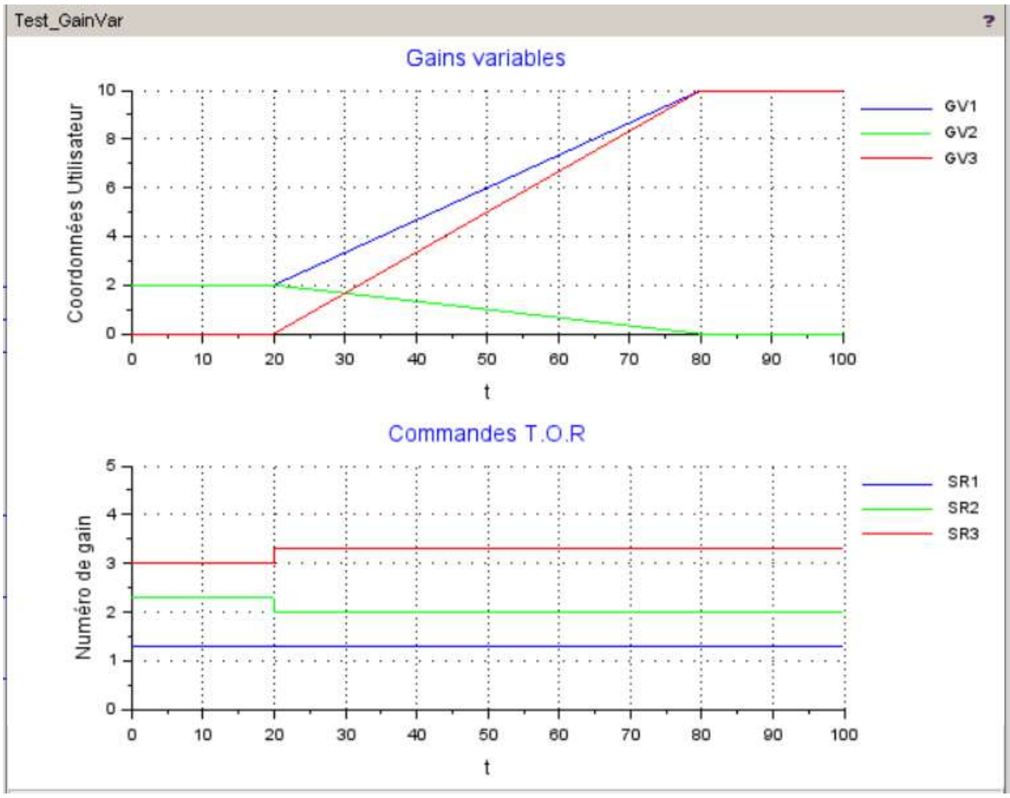
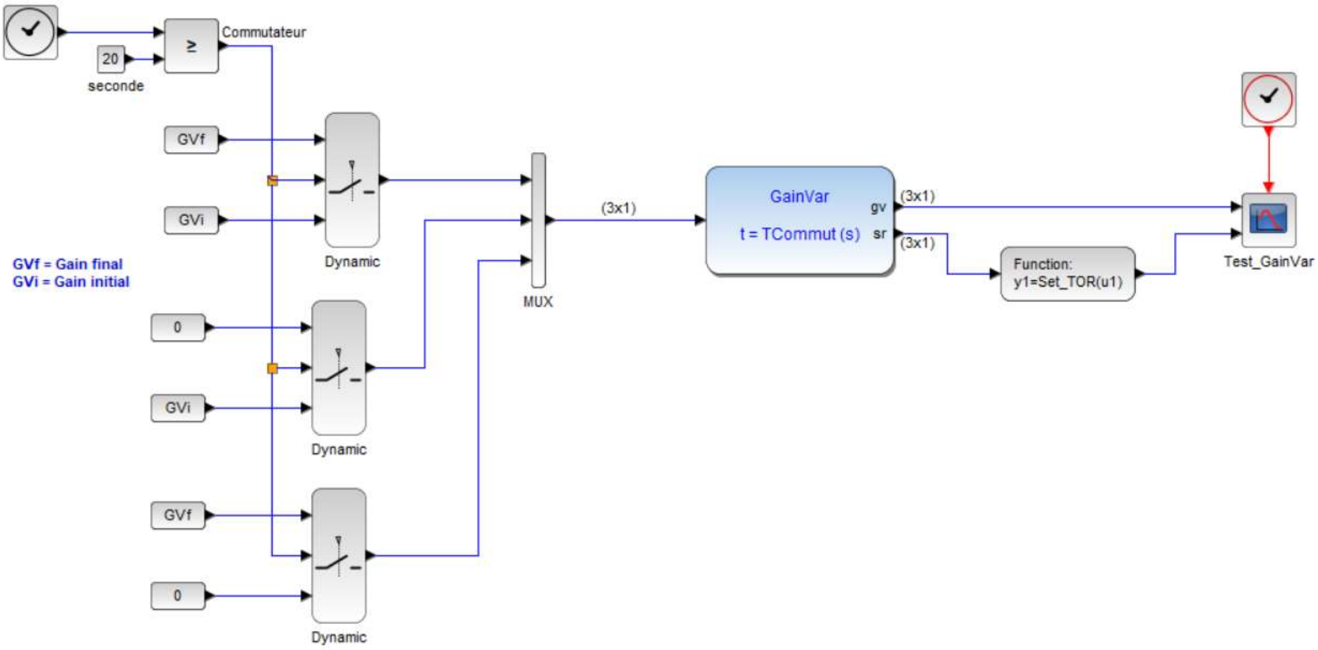


Figure 25 : GainVar.zcos test program – Monovariable implementation



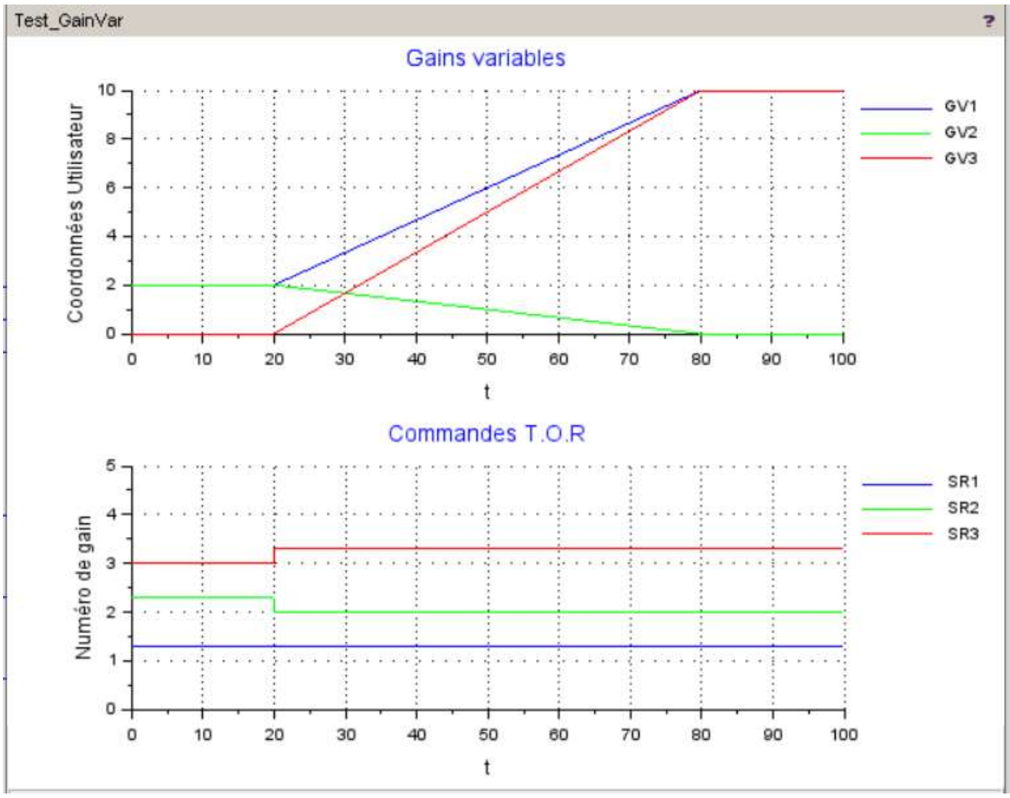


Figure 26 : Test program GainVar_V.zcos – Multivariable implementation

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.

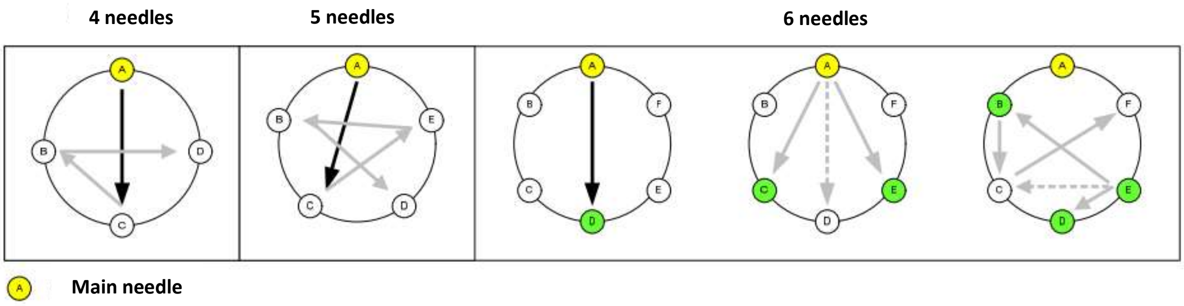


Figure 27 : Principle of jet switching, with respect to the calculations of radial forces on the PELTON wheel

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

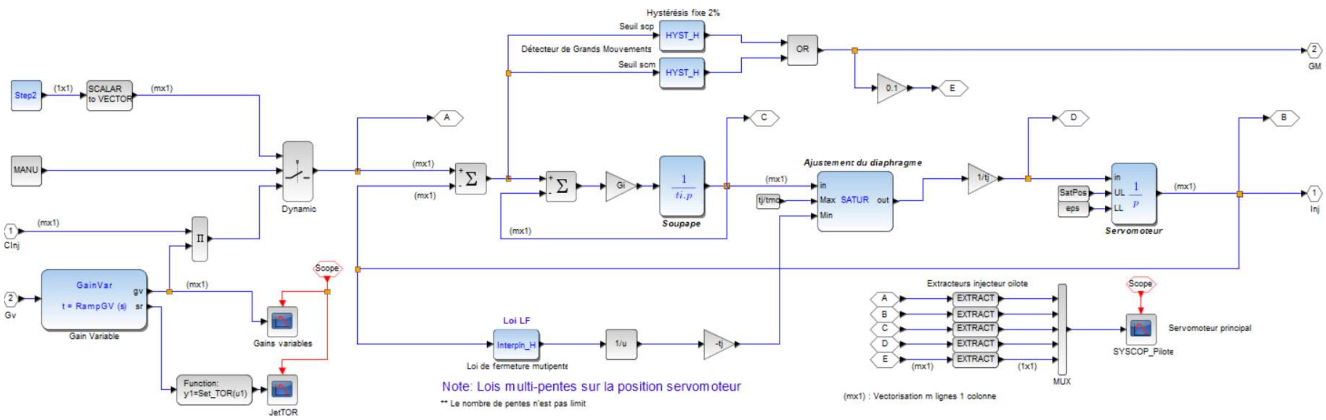


Figure 28 : Extract from systems copying SYSCOP_I of injectors (see PELTON diagram library) – Multivariable implementation

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.

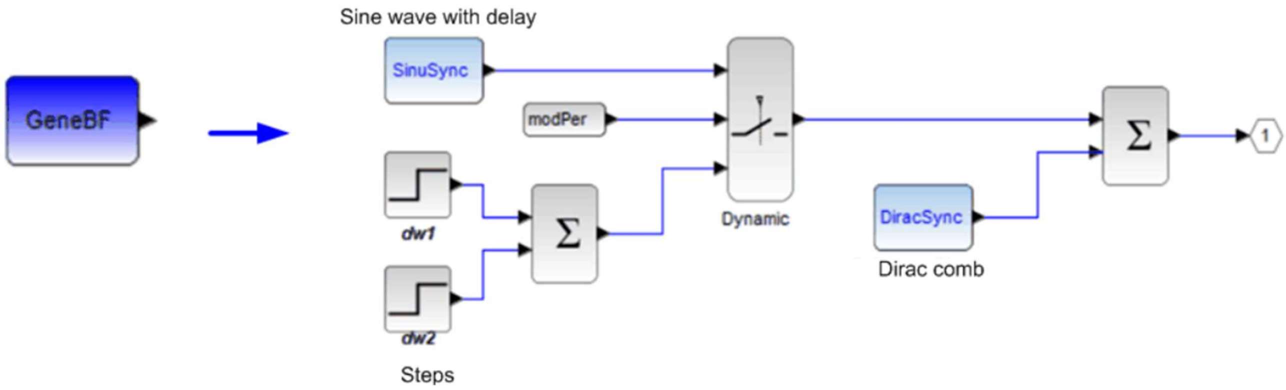


Figure 29 : Equivalence of the GeneBF generator with a SuperBlock

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

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

Parameters

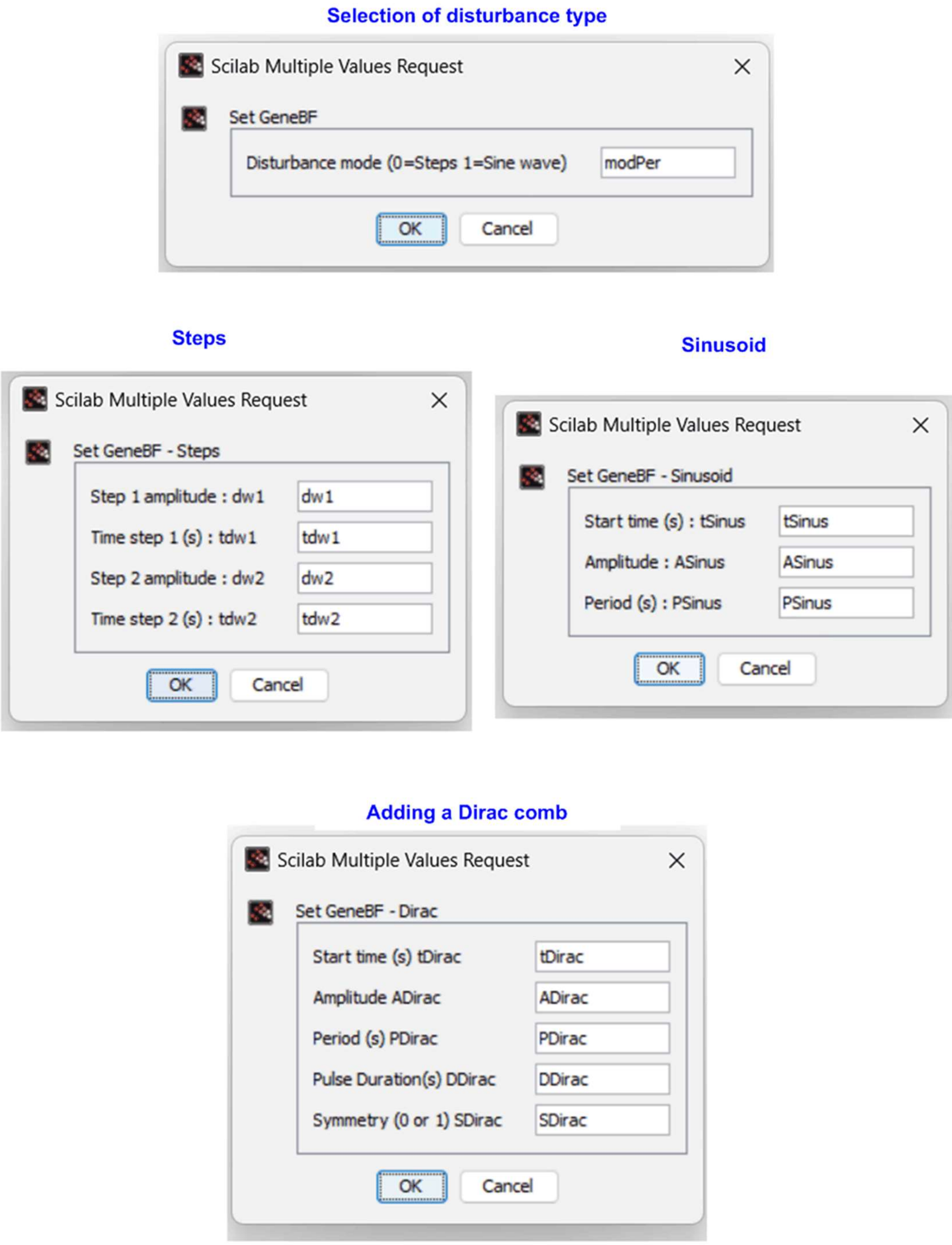


Figure 30 : GeneBF Generator Operator Interface

☞ **The reader will also consult:**

- **The technical note "NT2021.01.15 - HYDROTUR - BF Generator - Network Disturbance Recorders", located in the delivery directory "HYDROTUR\120 - HYDROTUR - Dissemination Journal - Technical Notes"**

- **The document "HYDROTUR - Programming Guide" located in the directory "HYDROTUR\3 - HYDROTUR - Programming Guide", chapter "GeneBF Power / Frequency Disturbance Generator"**

Alternative forms	-
Interfacing function	SEGPAL\macros\GeneBF.sci
Computational function	SEGPAL\src\c\GeneBFC.c
Test program	SEGPAL\demos\GeneBF.zcos

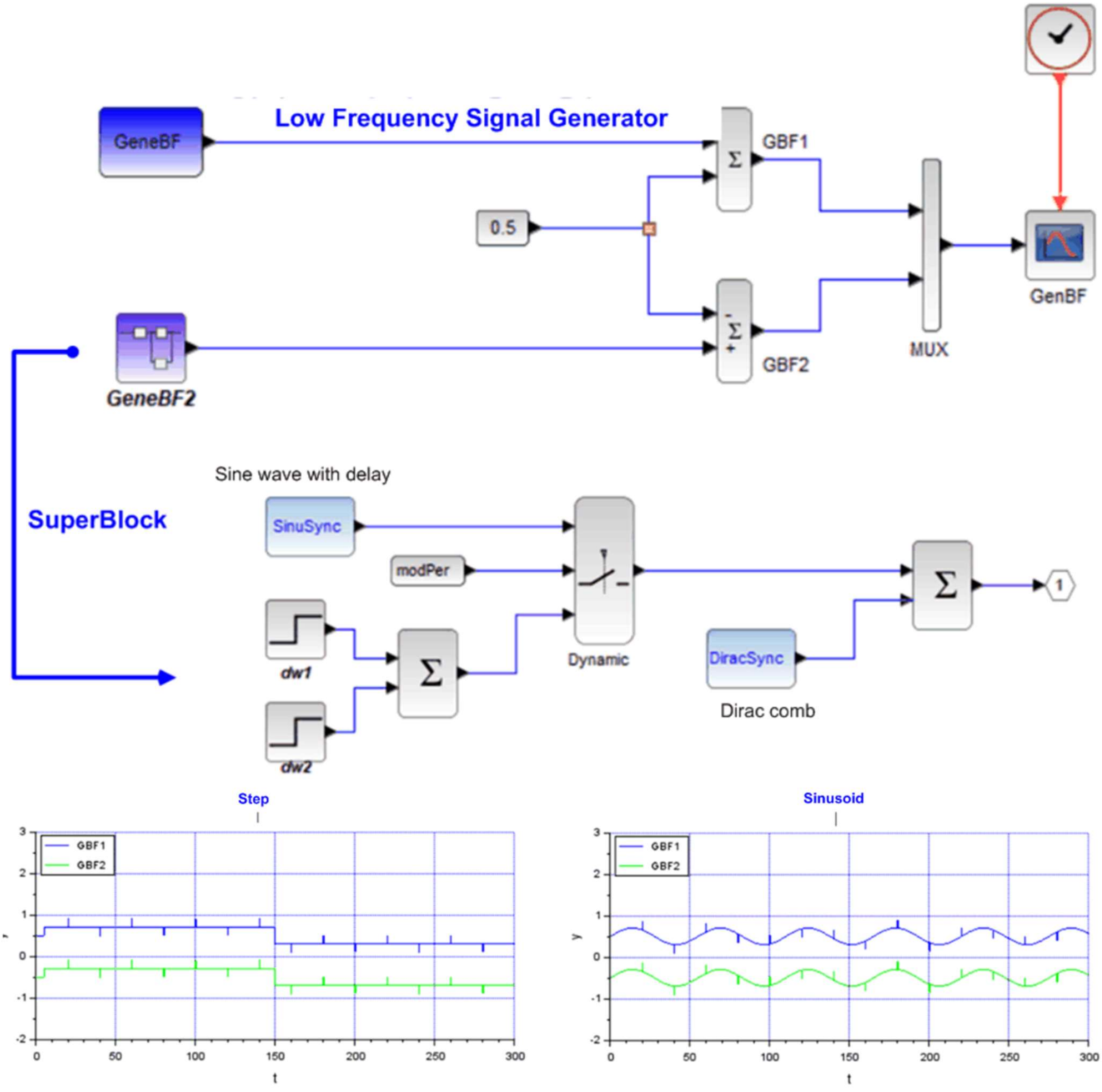


Figure 31 : GeneBF.zcos Generator Test Program

Notes

Use of the GeneBF disruptor in the "Network" modules of HYDROTUR simulators

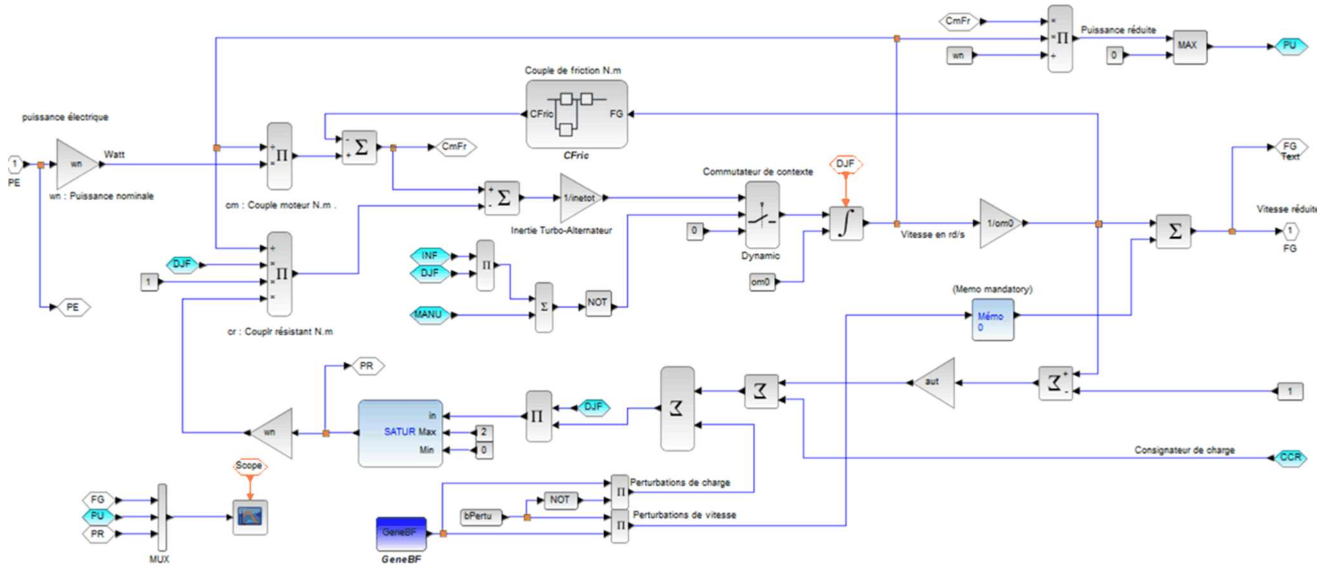
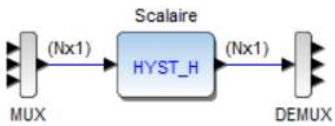


Figure 32 : Use of the GeneBF disruptor in the “Network” modules of HYDROTUR simulators

Hysteresis_H — Multivariable hysteresis

Block Screenshot

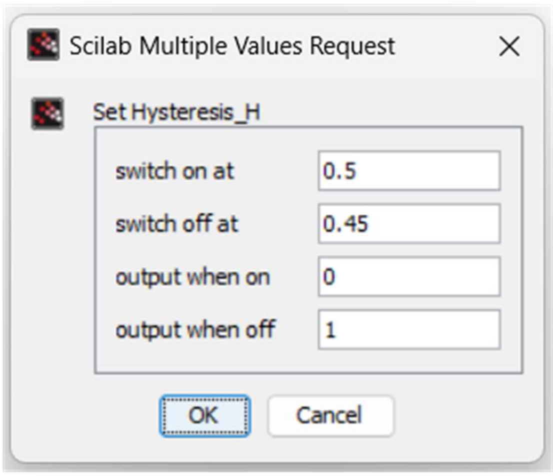


Description

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 "Switch on at" parameter. The block accepts a multivariable Nx1 input 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

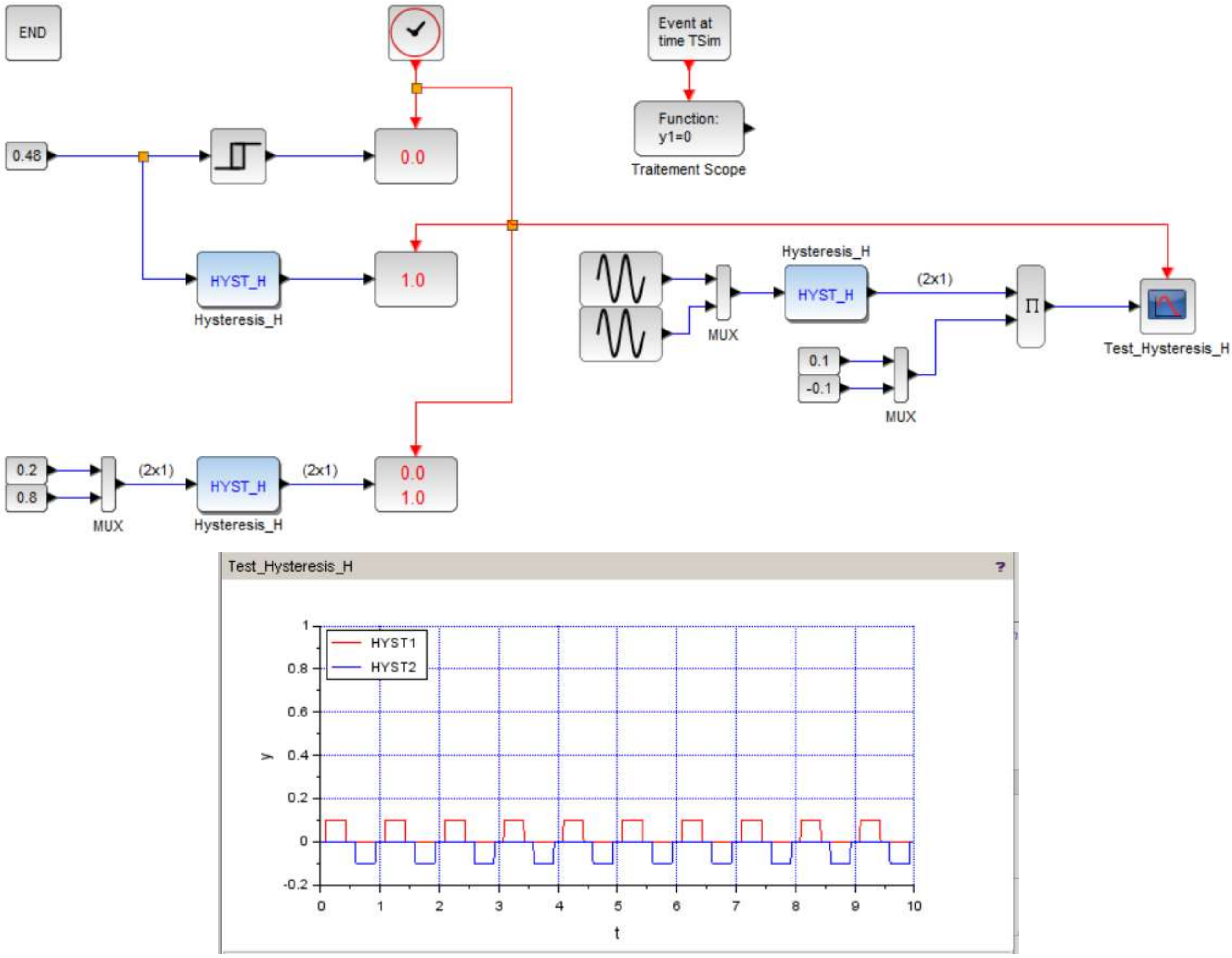


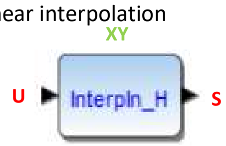
Figure 33 : Hysteresis_H.zcos test program

Notes ☞ See also *Figure 28 : – See also* ***Relais_H****.*

Interpln_H

Multivariable linear interpolation

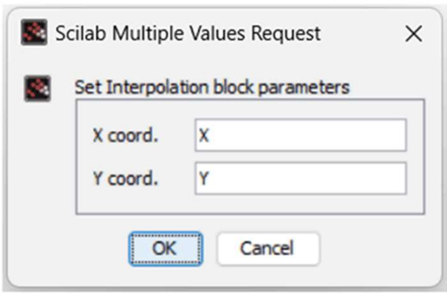
Block Screenshot



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

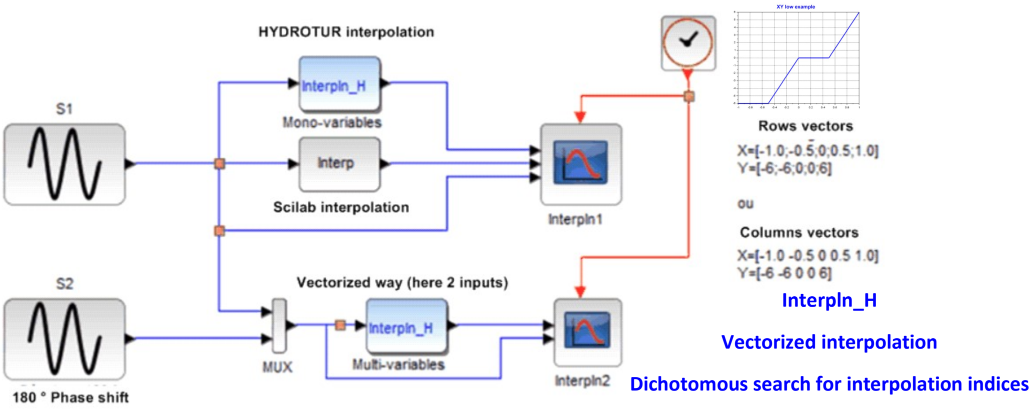


The vectors X and Y can be either 2 row vectors or 2 column vectors of the same dimension.

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



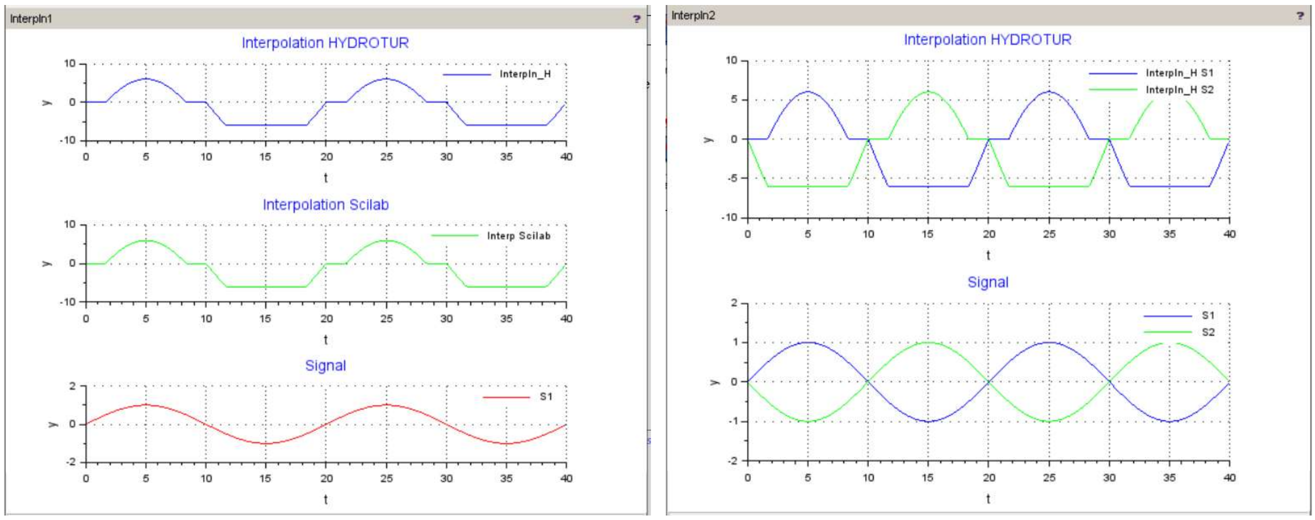
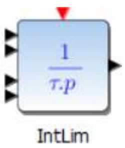


Figure 34 : Interpln_H.zcos test program

IntLim — Pure integrator with reset and external saturations

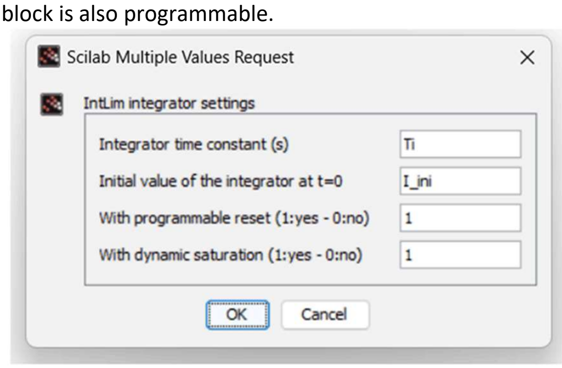
Block Screenshot



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

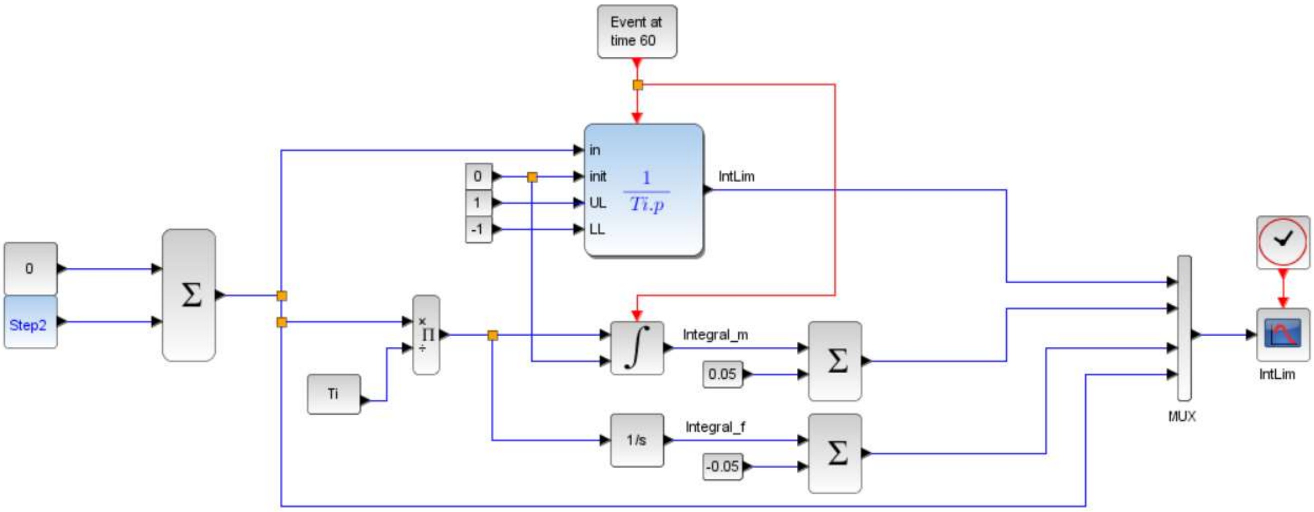


Alternative forms

Interfacing function SEGPAL\macros\IntLim.sci

Computational function SEGPAL\scr\c\IntLimC.c

Test program SEGPAL\demos\IntLim.zcos



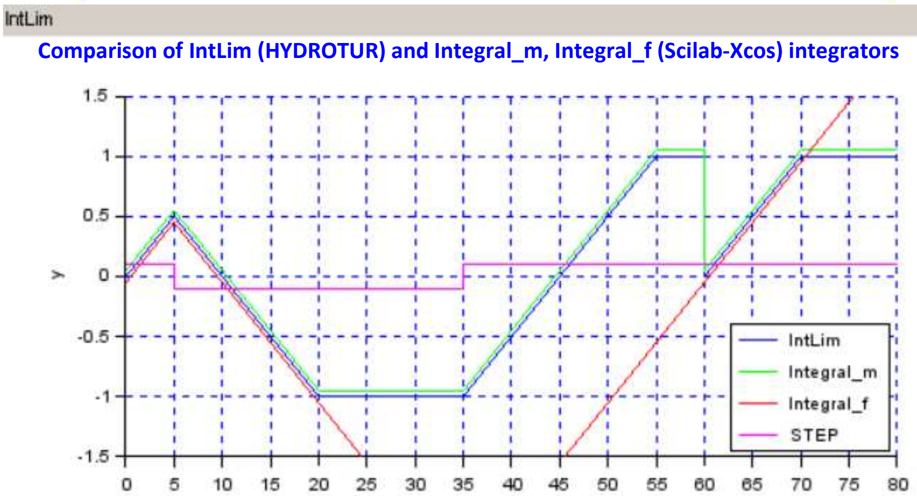


Figure 35 : IntLim.zcos test program

Notes The IntLim integrator is mainly used n the REGxx PID blocks of HYDROTUR speed governor.

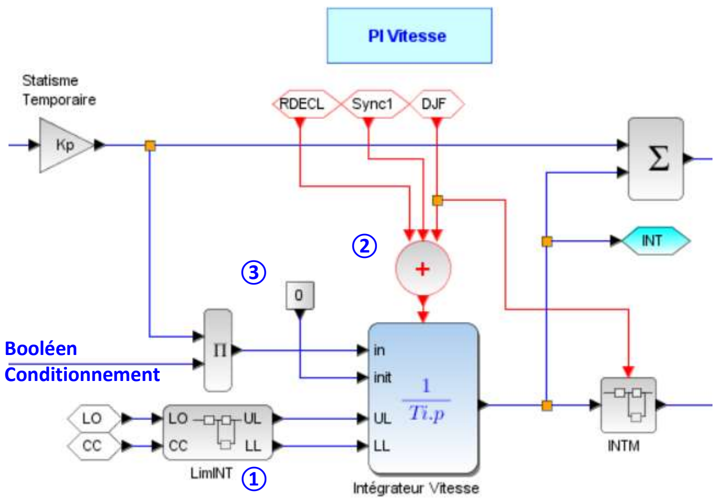


Figure 36 : IntLim integrator in the HYDROTUR speed governor

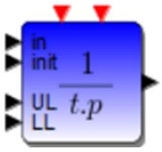
This includes, in this version:

- 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** : Unit trip relay
  - ✓ **DJF** : Closing the unit circuit breaker
  - ✓ **Sync1** : Resynchronization on significant speed difference

IntSPE

Special integrator with reset, external saturations, and fast desaturation

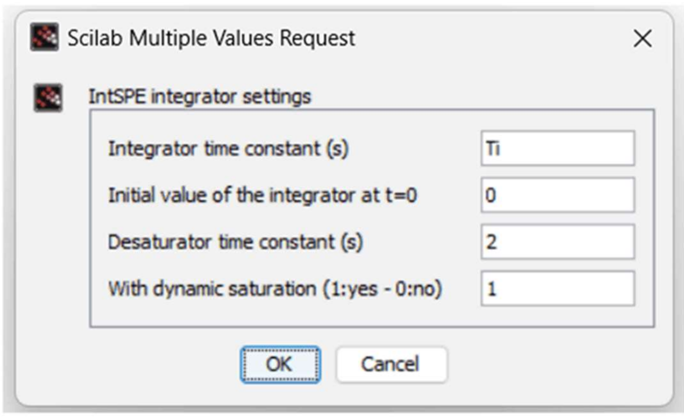
Block Screenshot



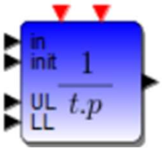
Description

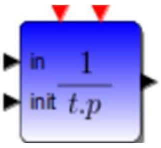
The IntSPE block integrates its input signal. It includes, as an option, saturation control devices, or event reset.
The initial value of the block is also programmable.
The block differs from the "**IntLim**" block by the possibility of event-driven desaturation. The desaturation time (reset to 0) is 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

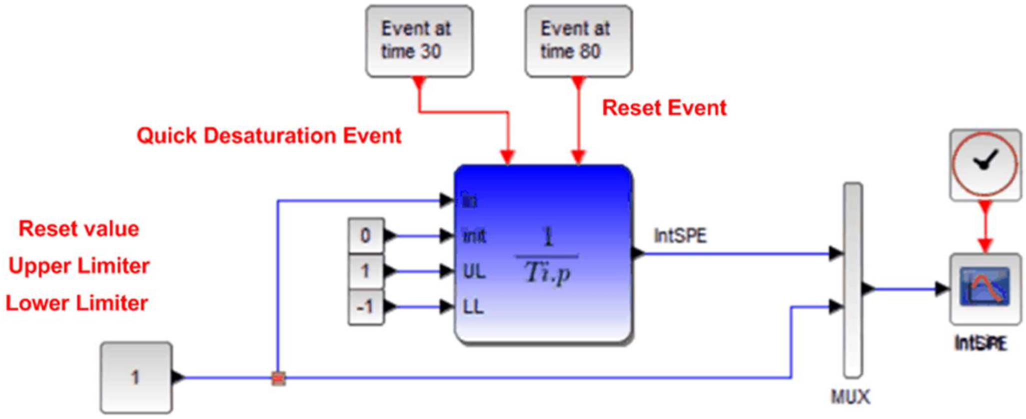


Figure 37 : Leaky integrator IntSPE

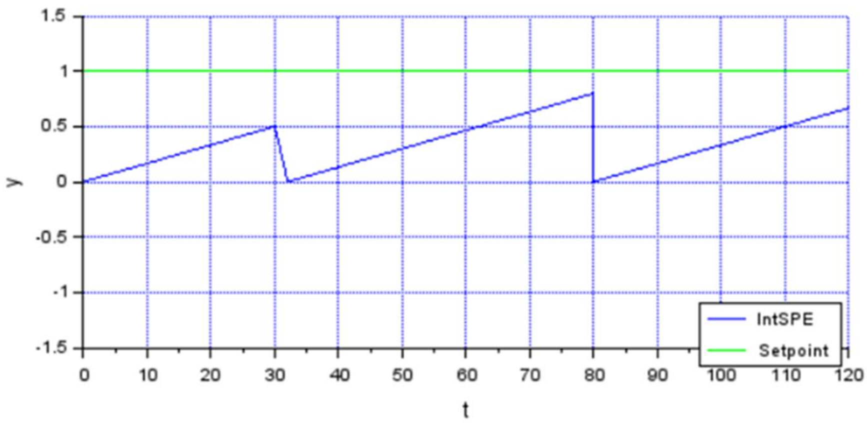


Figure 38 : IntSPE.zcos test program

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

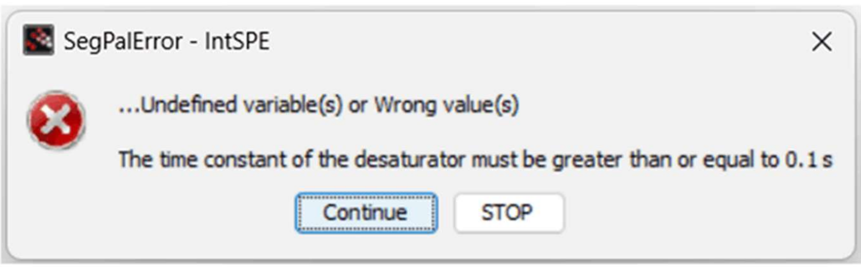
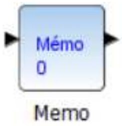


Figure 39 : InsSPE Integrator Desaturation Time Control

Memo

Pure delay

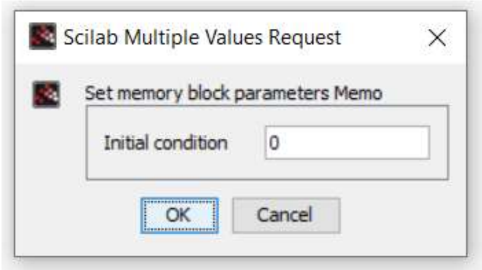
Block Screenshot



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

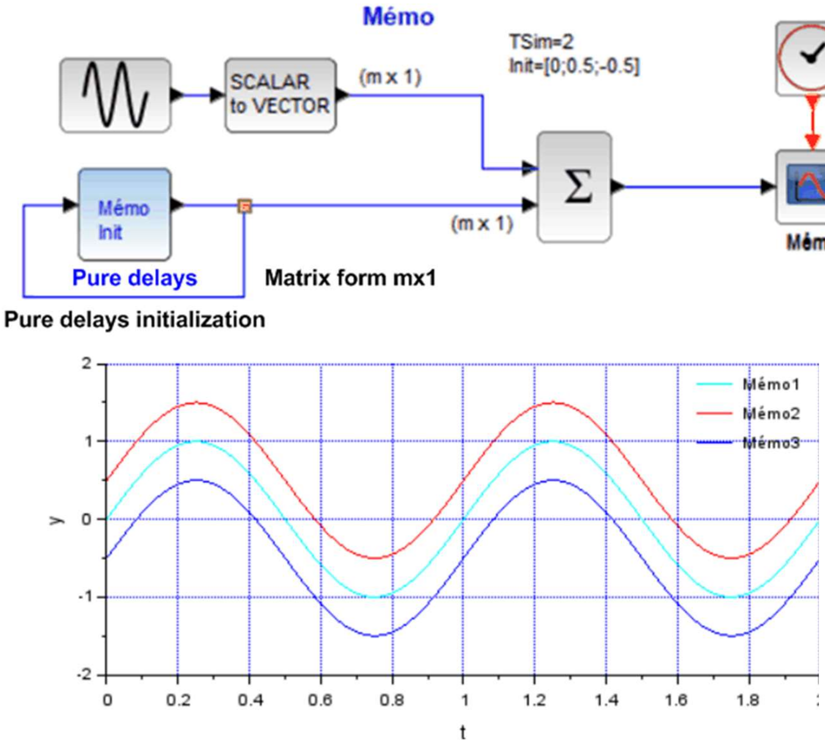


Figure 40 : Memo.zcos test program

Notes

The Memo block is used in the TURBADDUC block of the FR5 model, 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

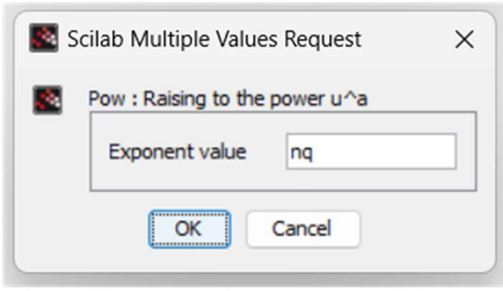
Raising a positive number to the power

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

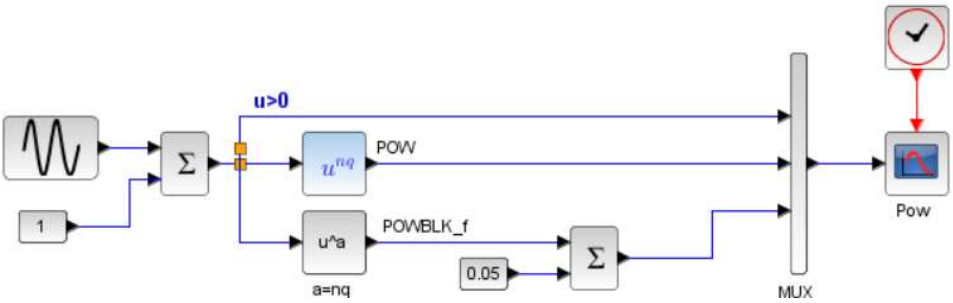


Alternative forms -

Interfacing function SEGPAL\macros\ Pow.sci

Computational function SEGPAL\scr\c\ ExponentC.c

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



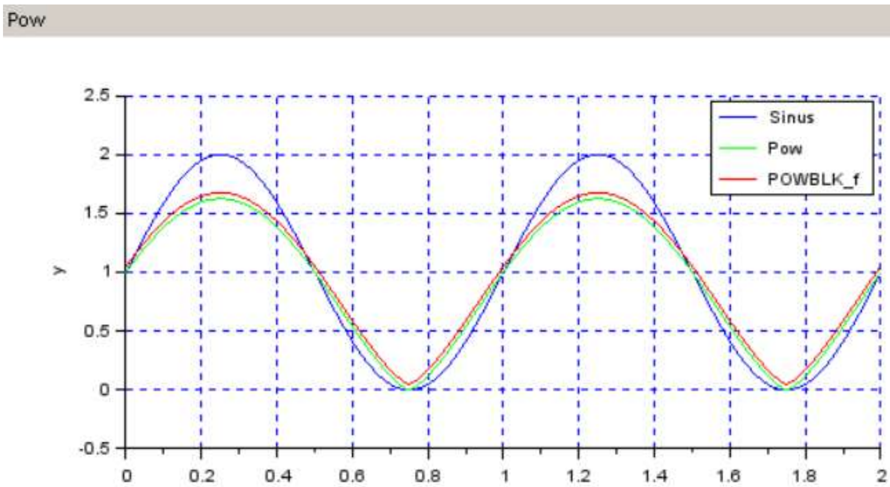


Figure 41 : Pow.zcos test program

Notes

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

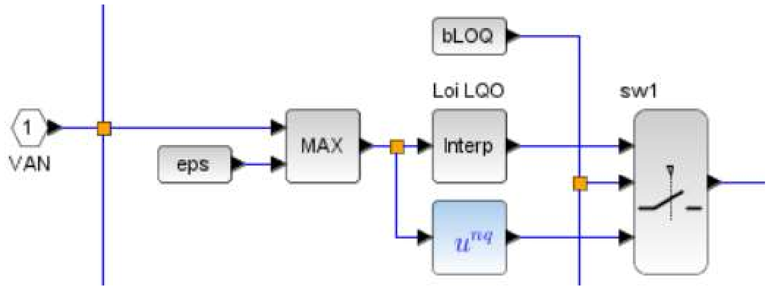
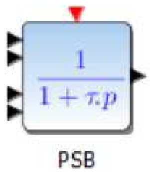


Figure 42 : Approximation of the Flow/Opening law in TURBADDUC model FR2

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

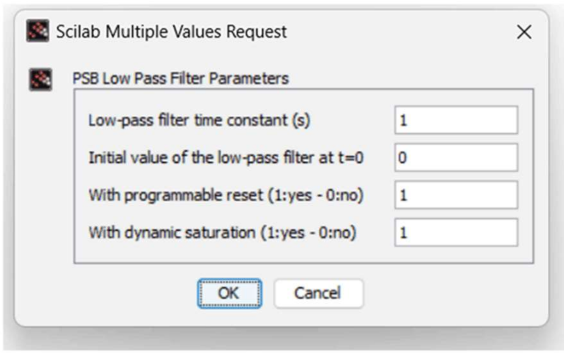
Block Screenshot



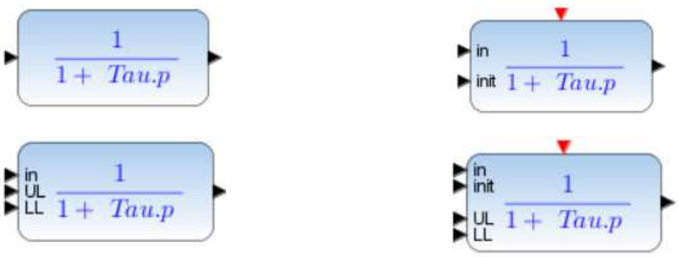
Description

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.

Parameters



Alternative forms



-

Interfacing function SEGPAL\macros\ PSB.sci

Computational function SEGPAL\scr\c\ PSBC.c

Test program SEGPAL\demos\ PSB.zcos

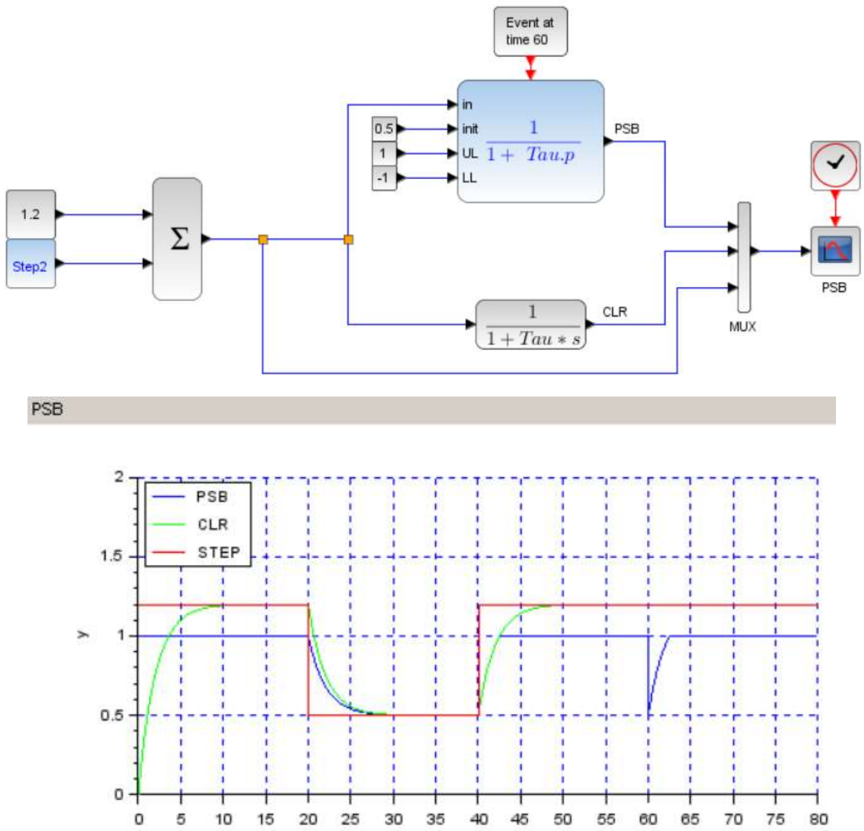


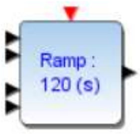
Figure 43 : PSB.zcos test program

Notes

Rampe

Integrator ramp with reset and external saturations

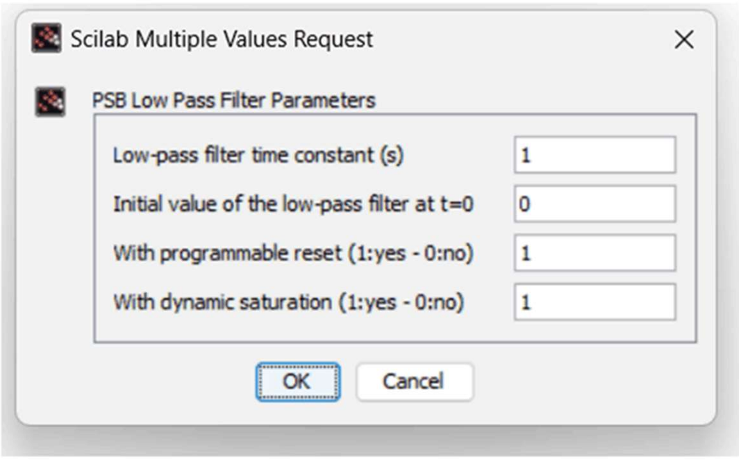
Block Screenshot



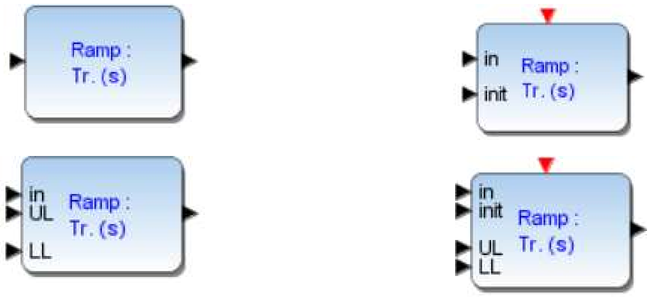
Description

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

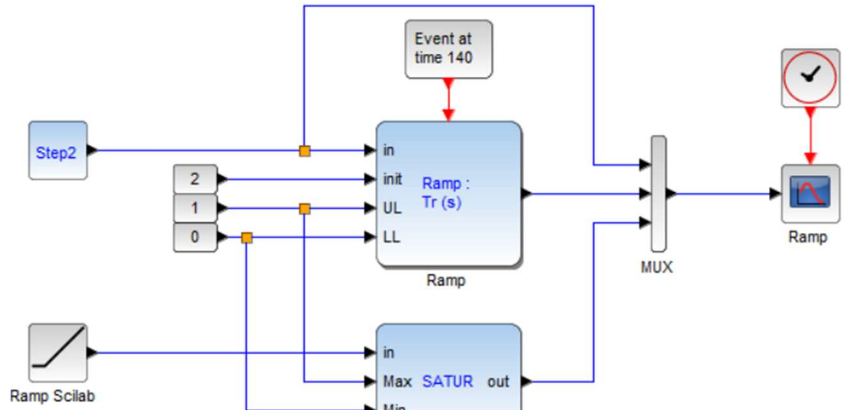


-

Interfacing function SEGPAL\macros\ Rampe.sci

Computational function SEGPAL\scr\c\ RampeC.c

Test program SEGPAL\demos\ Rampe.zcos



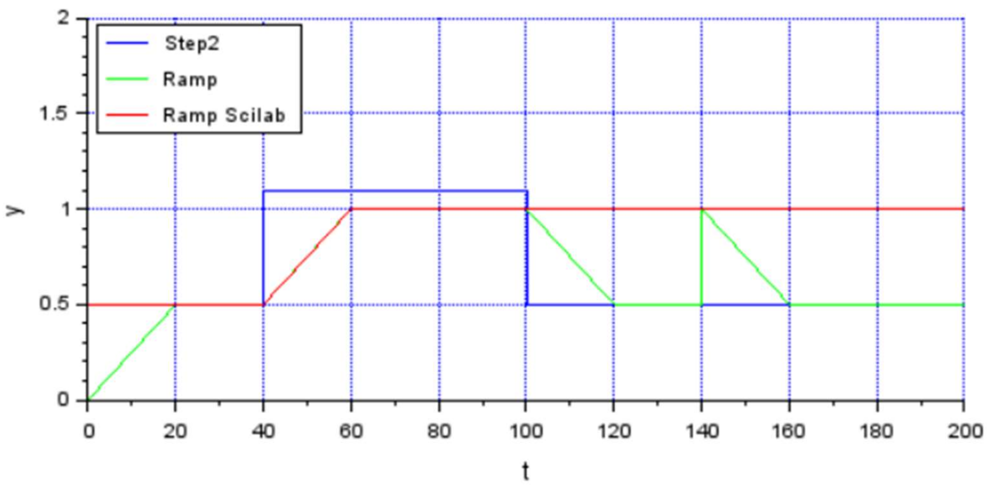


Figure 44 : Ramp.zcos test program

Notes

Figure 45 illustrates the use of the Ramp component in the setpoint loader of HYDROTUR 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 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.

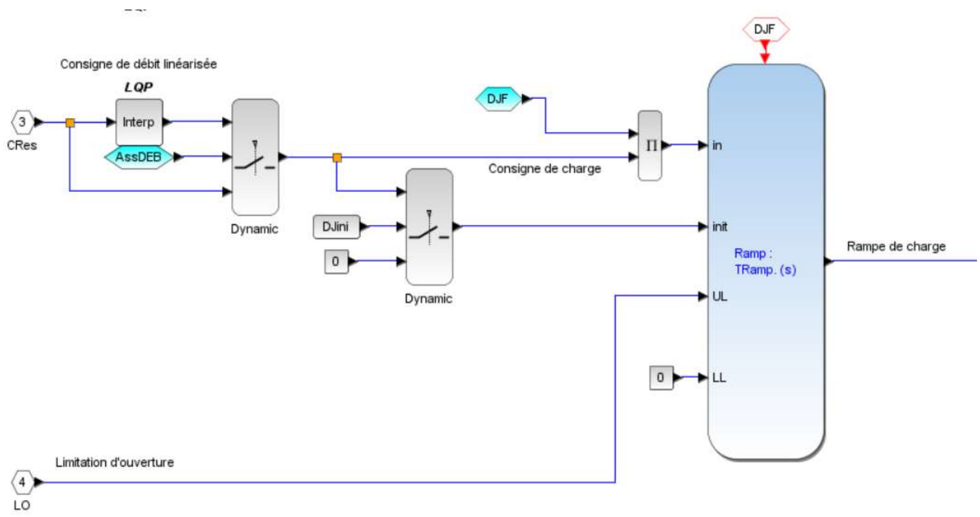


Figure 45 : Extract from the load recorder in HYDROTUR models

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:

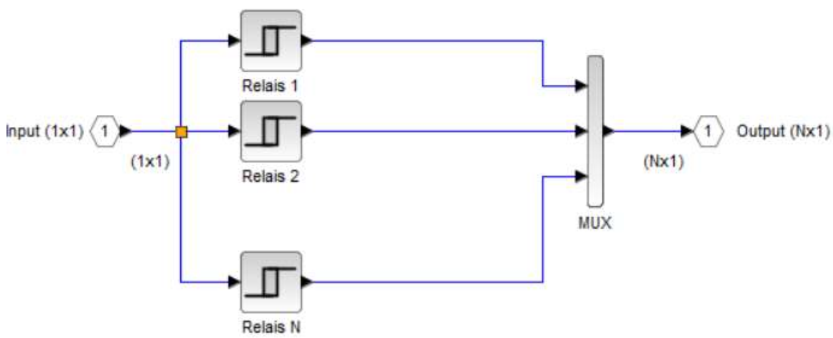
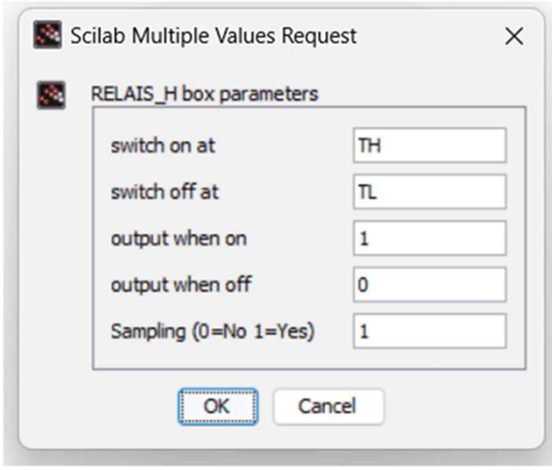


Figure 46 : Principle of the Relay_H component

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

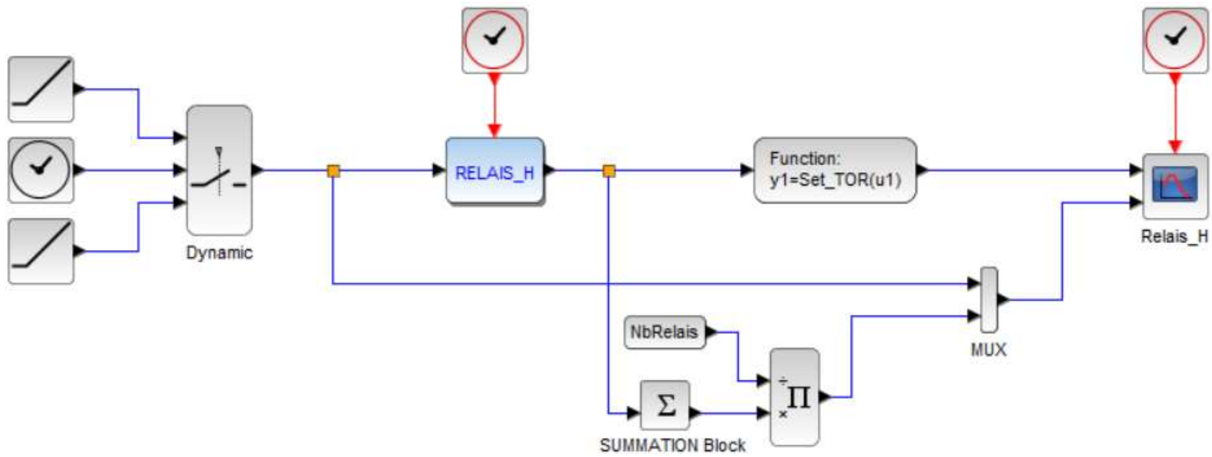
SEGPAL\macros\ Relais_H.sci

Computational function

SEGPAL\scr\c\ Relais_HC.c

Test program

SEGPAL\demos\ Relais_H.zcos, Répartiteur_Jets1.zcos, Répartiteur_Jets2.zcos



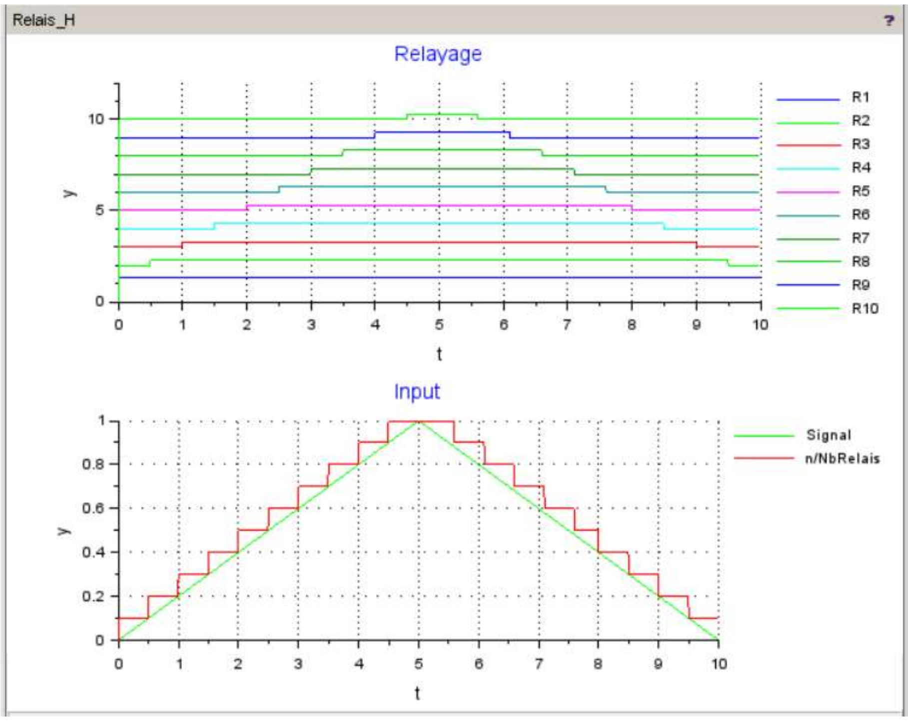
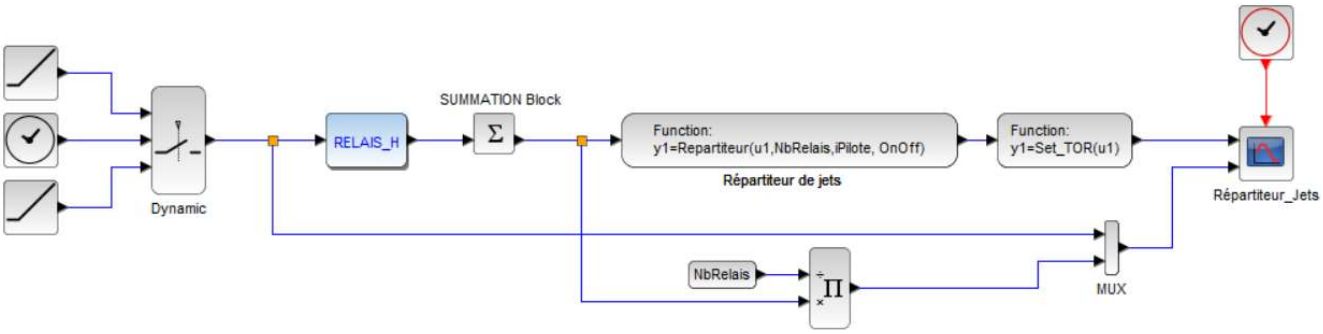


Figure 47 : Relay_H.zcos test program

Notes

See alsoi ***Hysteresis_H***

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



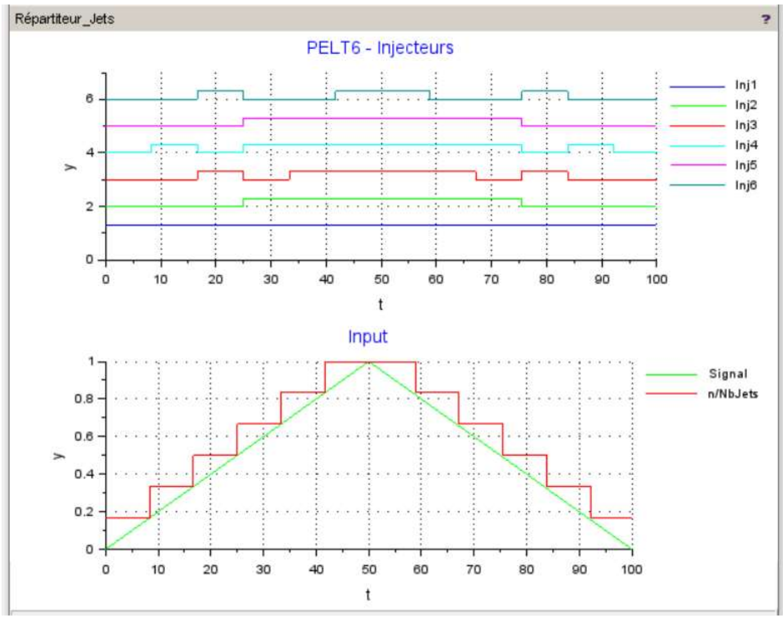


Figure 48 : Dispatcher_Jets1.zcos test program

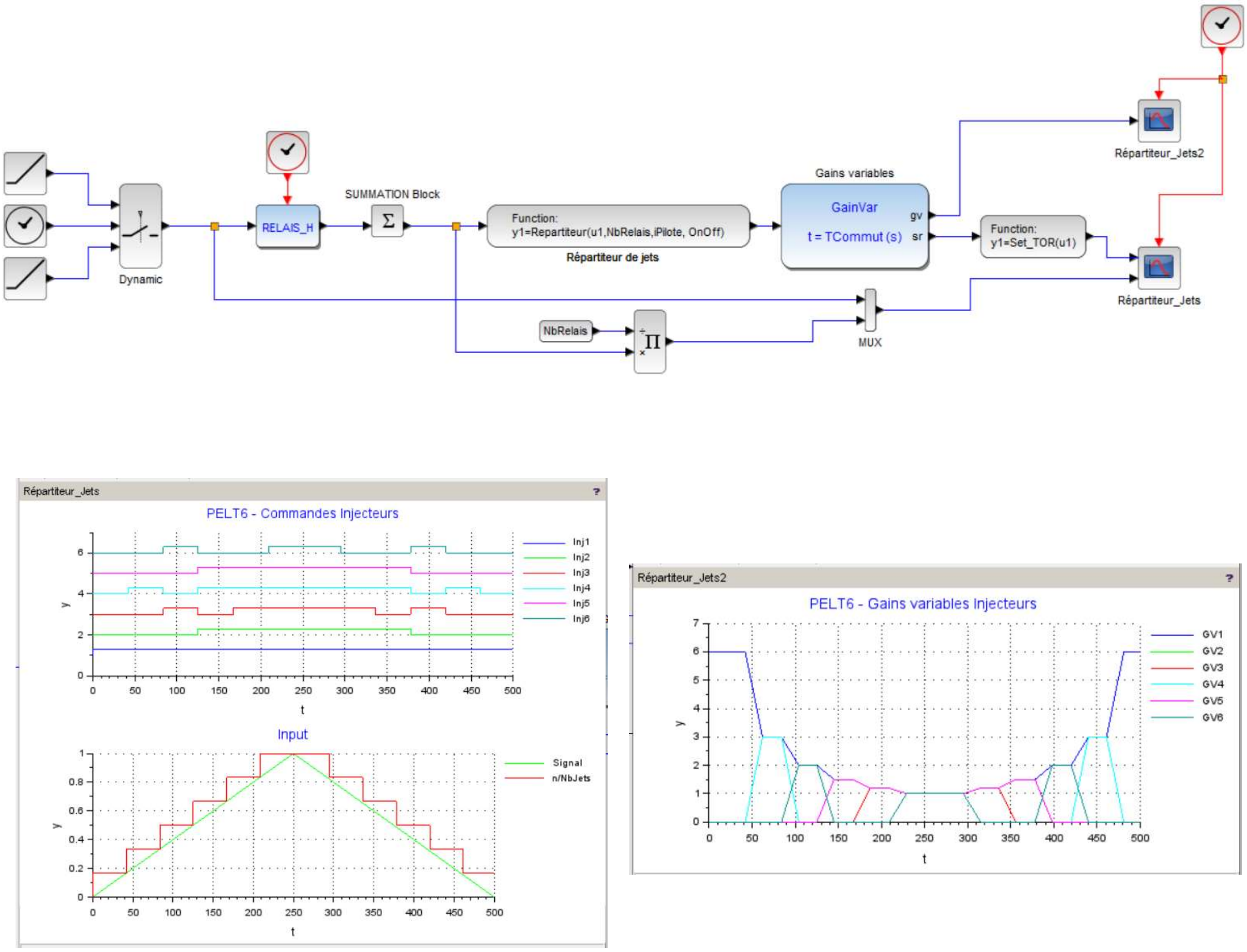


Figure 49 : Dispatcher_Jets2.zcos test program

RELATIONOP

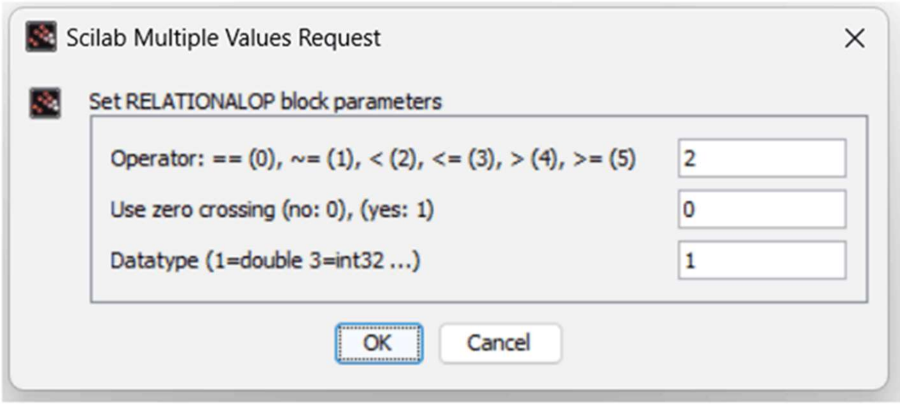
Multi-way relational operations

Block Screenshot

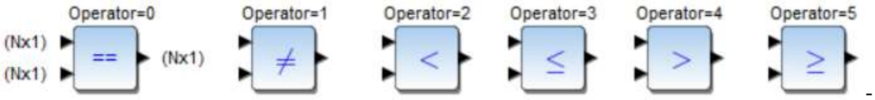
Description

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

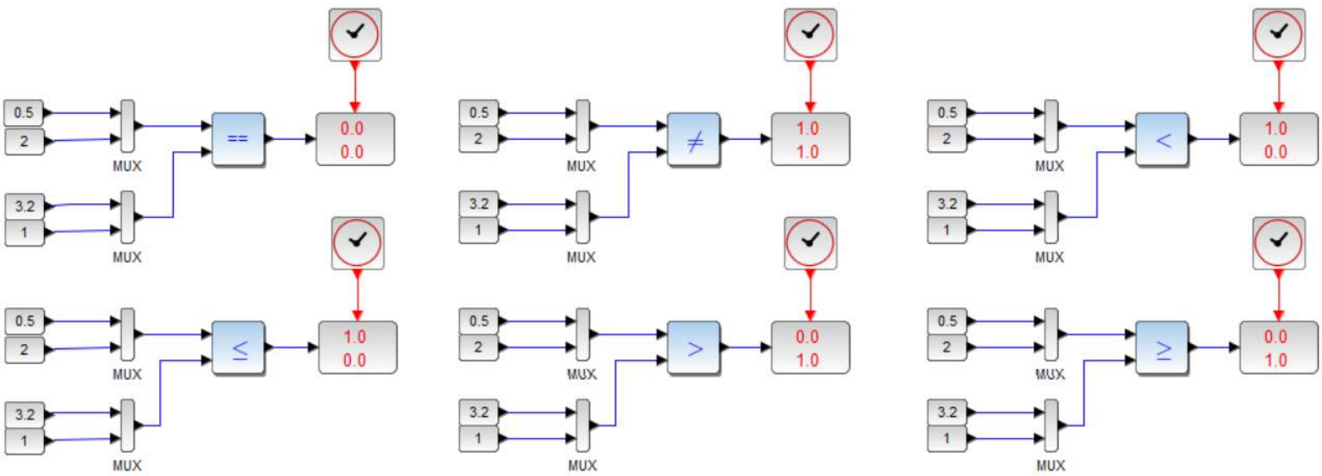


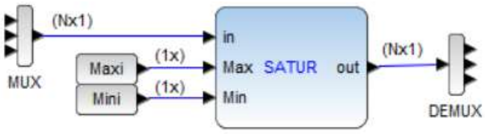
Figure 50 : RELATION_OP multivariables.zcos Test program

Notes

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

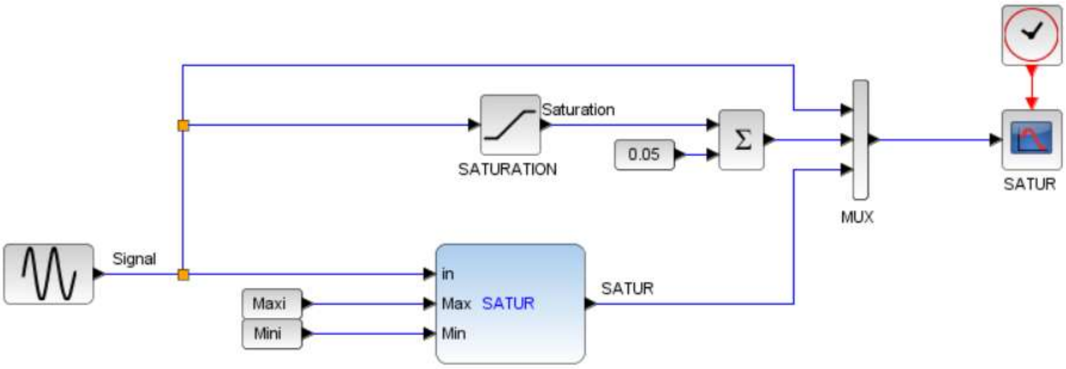
SEGPAL\macros\Satur.sci

Computational function

SEGPAL\scr\c\SaturC.c

Programme de test

SEGPAL\demos\Satur.zcos, Satur n lignes.zcos



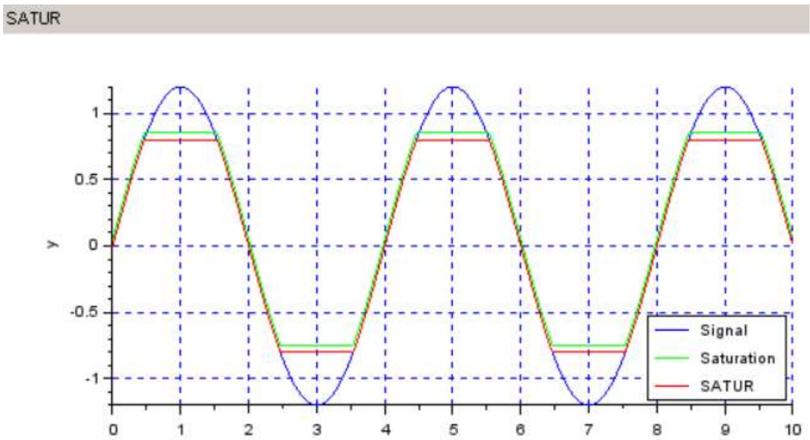


Figure 51 : Satur.zcos test program

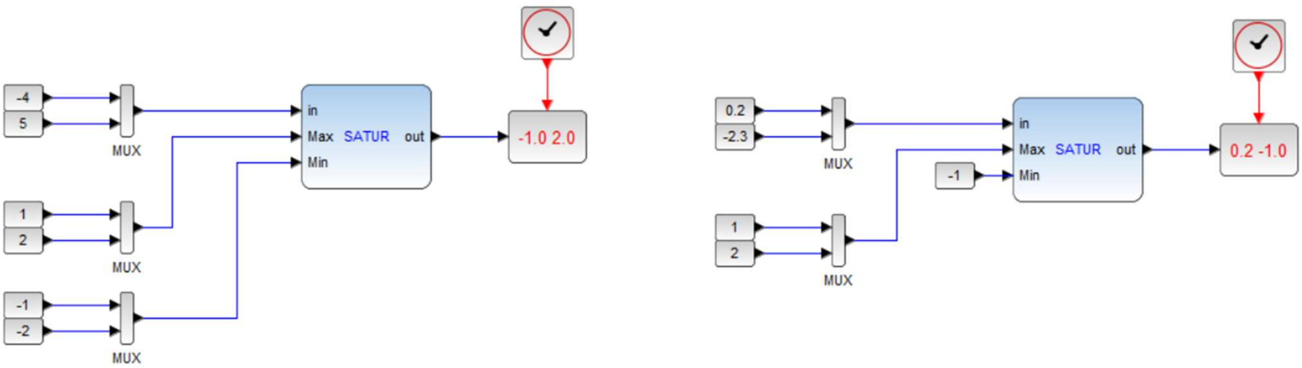


Figure 52 : Satur n lines.zcos test program

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

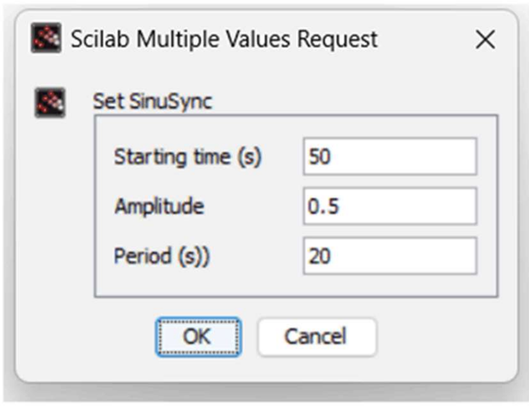
Description

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

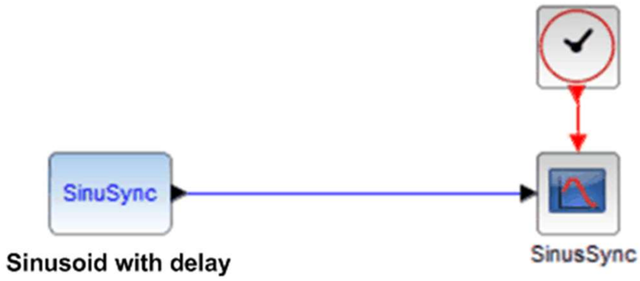


Alternative forms -

Interfacing function SEGPAL\macros\SinuSync.sci

Computational function SEGPAL\scr\c\SinuSyncC.c

Test program SEGPAL\demos\SinuSync.zcos
SEGPAL\demos\Sinus amorti.zcos



Also see GeneBF.zcos

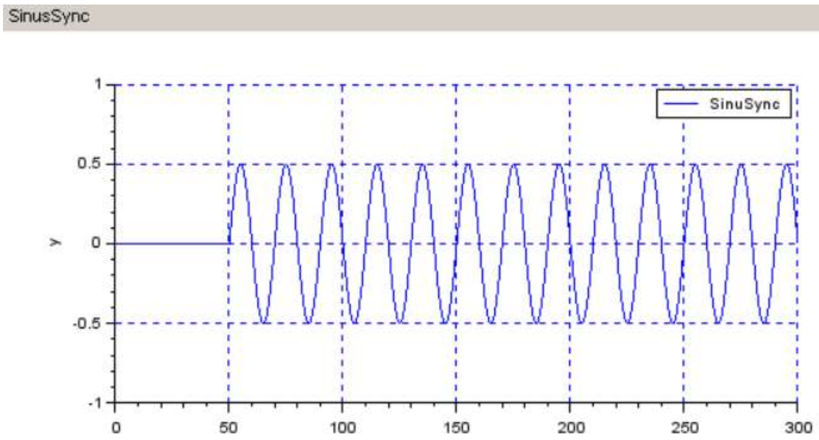


Figure 53 : SinuSync.zcos test program

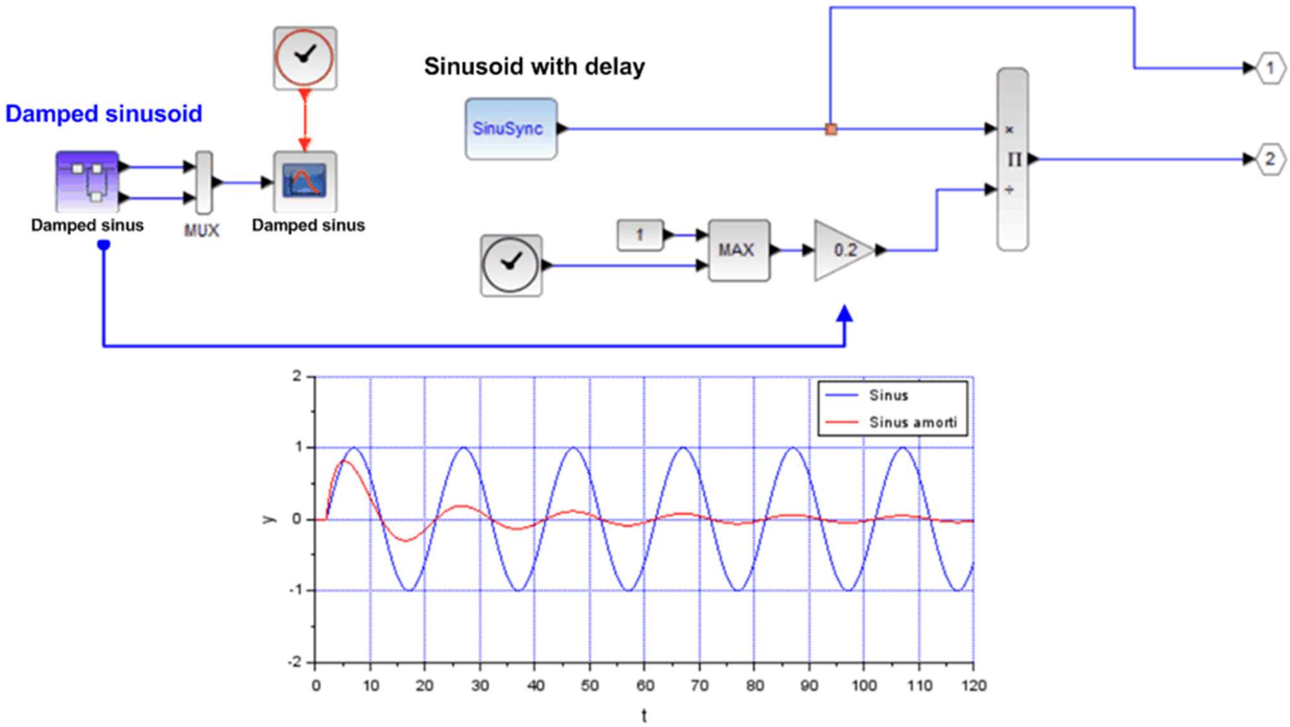


Figure 54 : Amorti.zcos test program

Notes

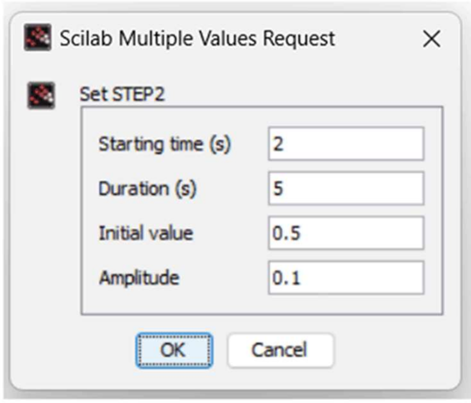
Step2 — Double programmable duration step

Block Screenshot

Description

The Step2 block generates two programmable steps from its interface.

Parameters

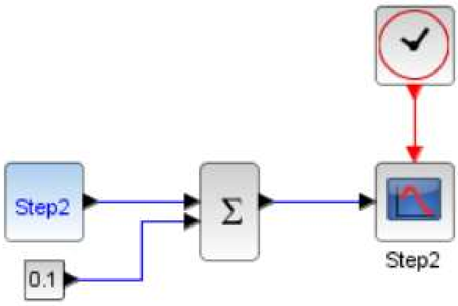


Alternative forms -

Interfacing function SEGPAL\macros\Step2.sci

Computational function SEGPAL\scr\c\Step2C.c

Test program SEGPAL\demos\Step2.zcos



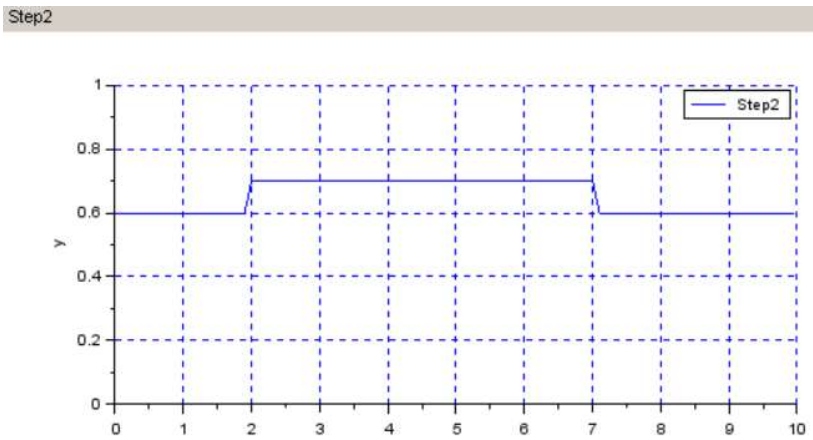


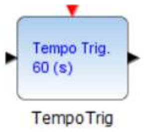
Figure 55 : Step2.zcos test program

Notes

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

TempoTrig — Retriggerable timer

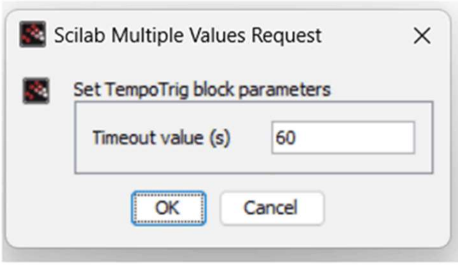
Block Screenshot



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, <u>if the in input is zero.</u>

Parameters

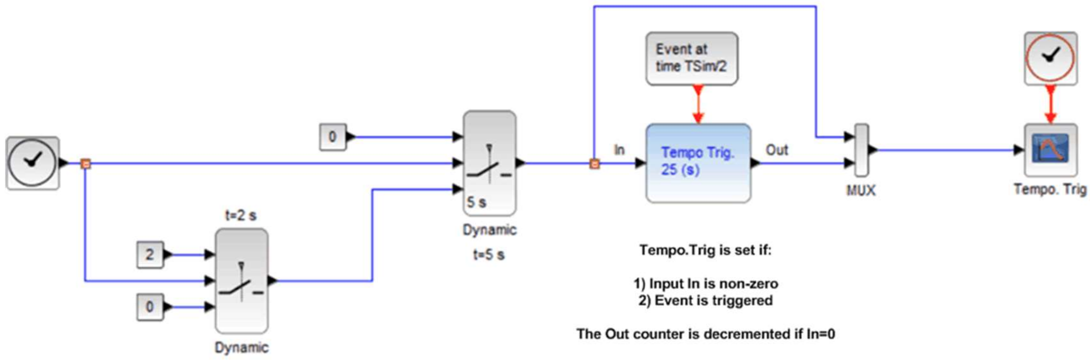


Alternative forms -

Interfacing function SEGPAL\macros\TempoTrig.sci

Computational function SEGPAL\scr\c\TempoTrigC.c

Test program SEGPAL\demos\TempoTrig.zcos



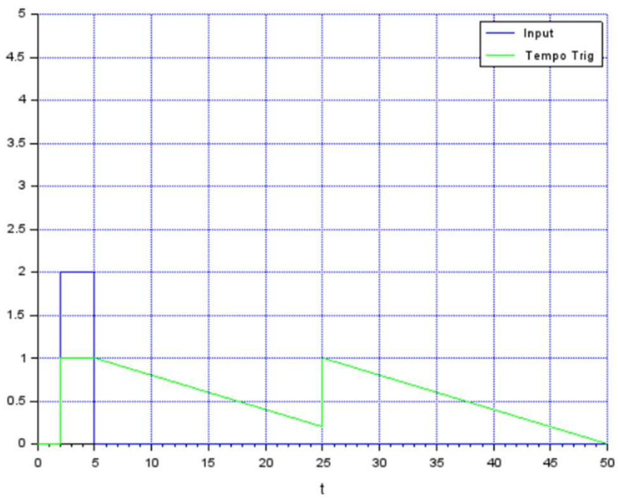


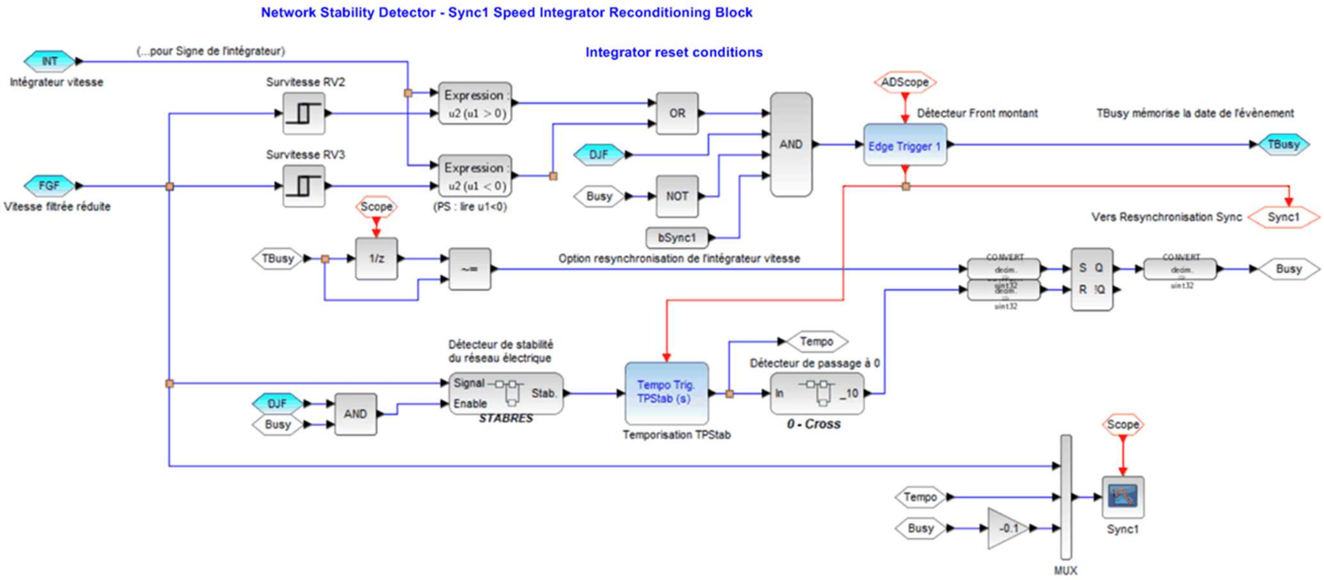
Figure 56 : TempoTrig.zcos test program

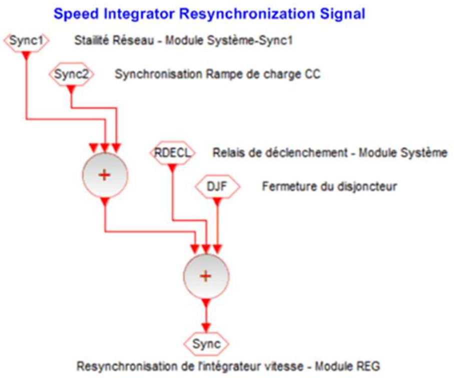
Notes

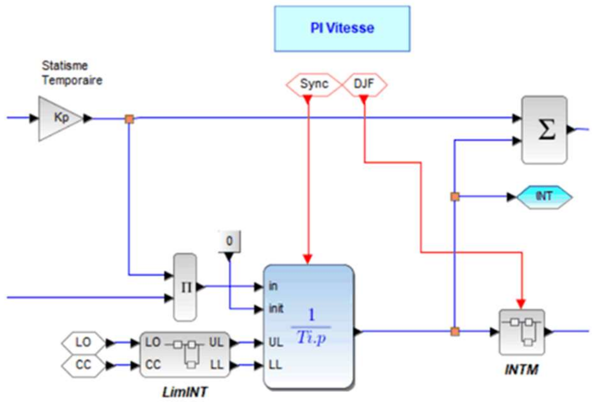
Also see EdgeTrigger, Busy.zcos Test program

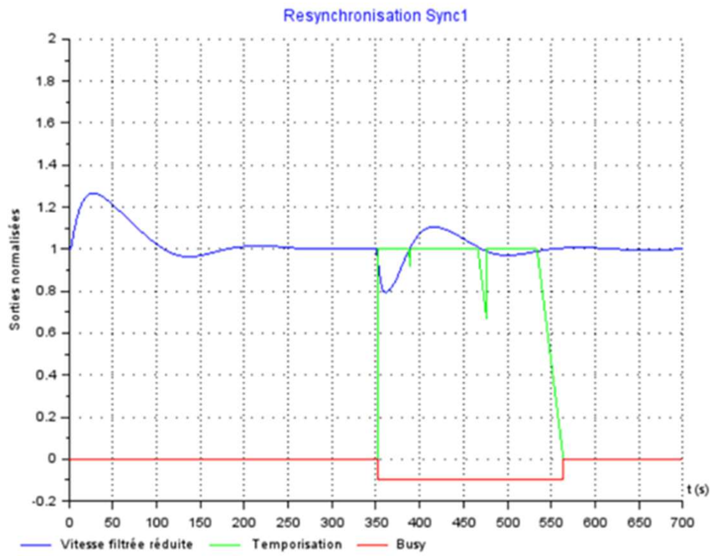
In some difficult cases (KAPLAN turbines Load steps at 75% of nominal power, 10% load steps), it may be necessary to reset the speed integrator by monitoring the evolution of the speed curve using a specialized SYNC1 resynchronization circuit.

:





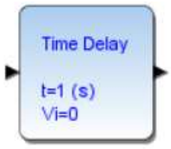




Time Delay

Input time offset

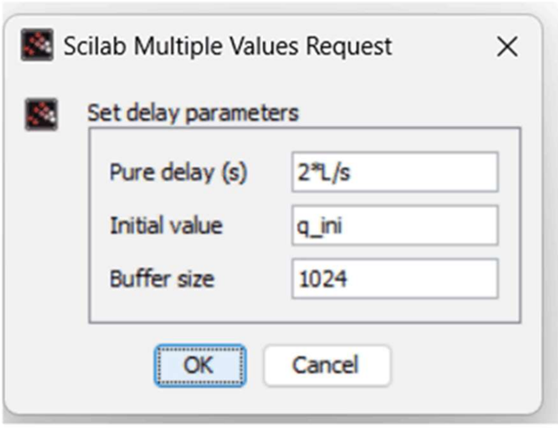
Block Screenshot



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

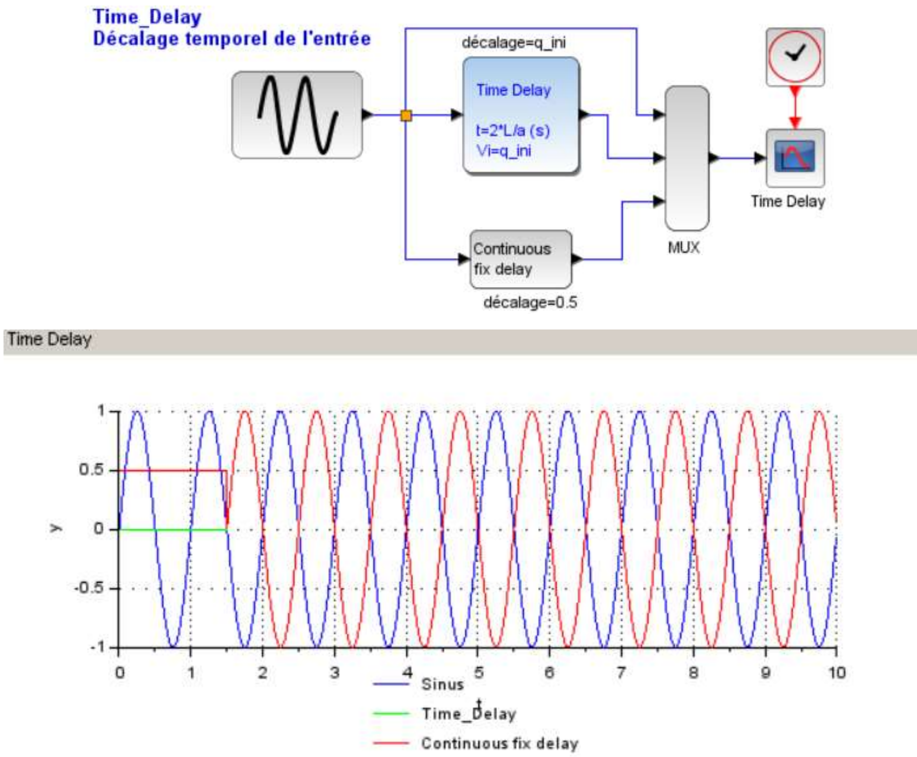


Figure 57 : TimeDelay.zcos test program

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 similar to 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 58 : Error reporting if Time Delay buffer size is not suitable

APPENDIX

A. TYPICAL LIST AFTER EXECUTION OF THE BUILDER.SCE BUILDER

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

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

Building macros...
-- Creation of [SegPallib] (Macros) --
genlib: Processing file: CLRseg.sci
genlib: Processing file: CheckExprs.sci
genlib: Processing file: DerFil.sci
genlib: Processing file: DiracSync.sci
genlib: Processing file: EdgeTrigger.sci
genlib: Processing file: GainVar.sci
genlib: Processing file: GeneBF.sci
genlib: Processing file: Hysteresis_H.sci
genlib: Processing file: IntLim.sci
genlib: Processing file: IntSPE.sci
genlib: Processing file: Interpln_H.sci
genlib: Processing file: Memo.sci
genlib: Processing file: PSB.sci
genlib: Processing file: Pow.sci
genlib: Processing file: RELATIONALOP_H.sci
genlib: Processing file: Rampe.sci
genlib: Processing file: Relais_H.sci
genlib: Processing file: Repartiteur.sci
genlib: Processing file: Satur.sci
genlib: Processing file: SegPalError.sci
genlib: Processing file: Set_TOR.sci
genlib: Processing file: SinuSync.sci
genlib: Processing file: Step2.sci
genlib: Processing file: TempoTrig.sci
genlib: Processing file: Time_Delay.sci
Building blocks...
Building sources...
  "...Compilateur C"
   Generate a loader file
   Generate a Makefile
   Running the Makefile
   Compilation of PSBC.obj
   Compilation of ExponentC.obj
   Compilation of IntLimC.obj
   Compilation of RampeC.obj
   Compilation of MemoC.obj
   Compilation of SaturC.obj
   Compilation of GainVarC.obj
   Compilation of Step2C.obj
   Compilation of EdgeTriggerC.obj
   Compilation of TempoTrigC.obj
   Compilation of SinuSyncC.obj
   Compilation of DiracSyncC.obj
   Compilation of Hysteresis_HC.obj
   Compilation of Relais_HC.obj
   Compilation of Interpln_HC.obj
   Compilation of Common.obj
   Compilation of IntSPEC.obj
   Compilation of GeneBFC.obj
   Building shared library (be patient)
   Generate a cleaner file
Generating cleaner_src.sce...
Building help...
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
Generating loader.sce...
Generating unloader.sce...
Generating cleaner.sce...
```

Figure 59 : Listing after running SegPal builder.sce builder

B. ELEMENTS OF IMPLEMENTATION OF TEST PROGRAMS

B.1.Setting simulation times

The overall duration of a simulation is set in the Xcos Setup menu in Scilab. This feature is accessible by accessing the **Simulation→ Setup** menu of XCOS, and sets a maximum simulation time duration of 10^5 s.

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

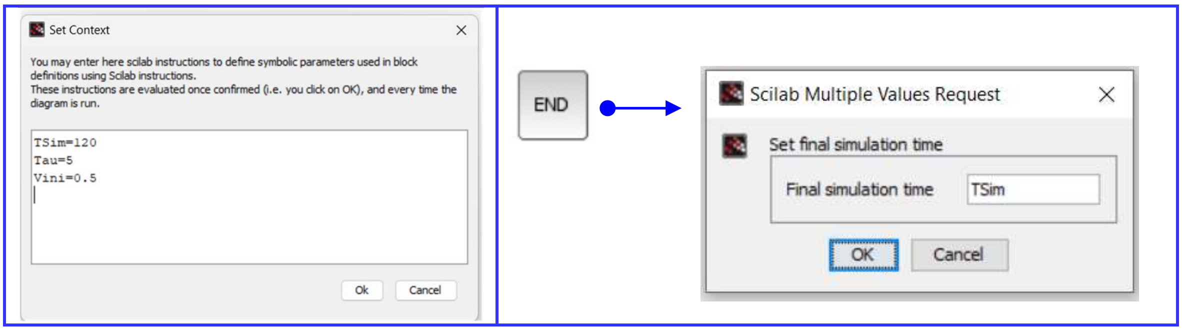


Figure 60 : Setting the TSim simulation time

B.2.Principle of numbering graphic windows

Each test program integrates a multiplexed graphics output:

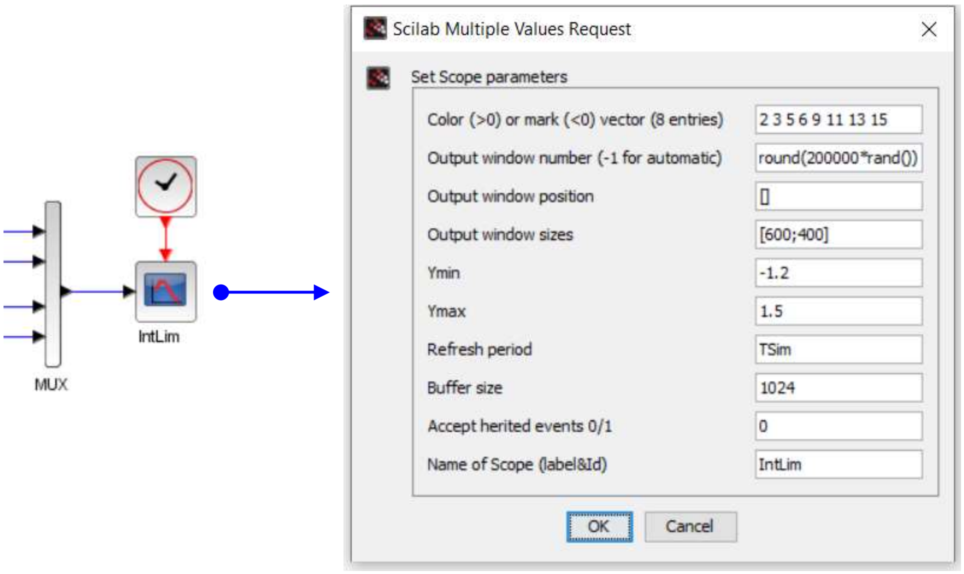


Figure 61 : Principle of numbering of graphic windows

In order to guarantee the uniqueness of each graphical 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 value closest to the floating number 200000*rand().

In fact, the "**-1 for Automatic**" advice does not guarantee the uniqueness of the window, when chaining several programs, or restarting the same program.

B.3. Managing legends associated with graphic windows

A legend is automatically inserted into the graphics window of a test program, each time a simulation is completed.

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

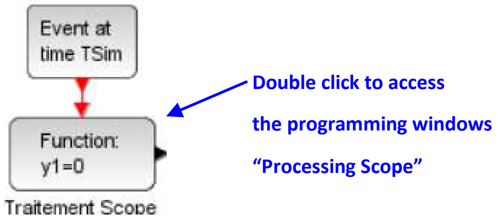
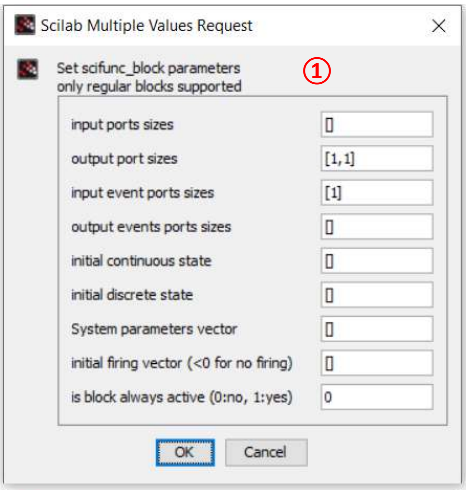
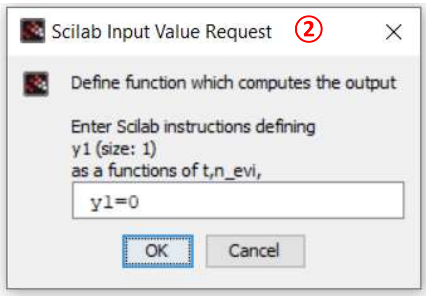
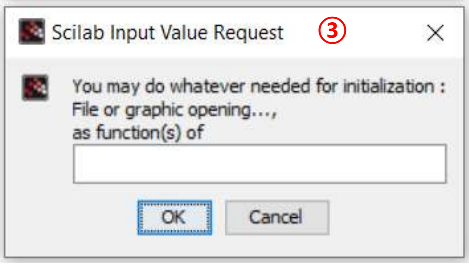


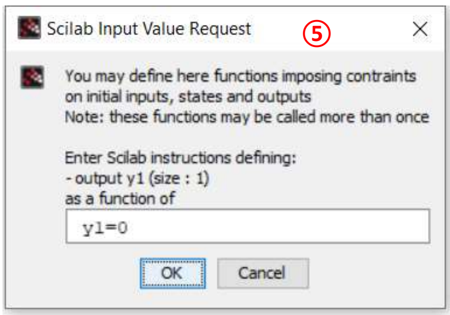
Figure 62 : Principle of legend insertion in test programs

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









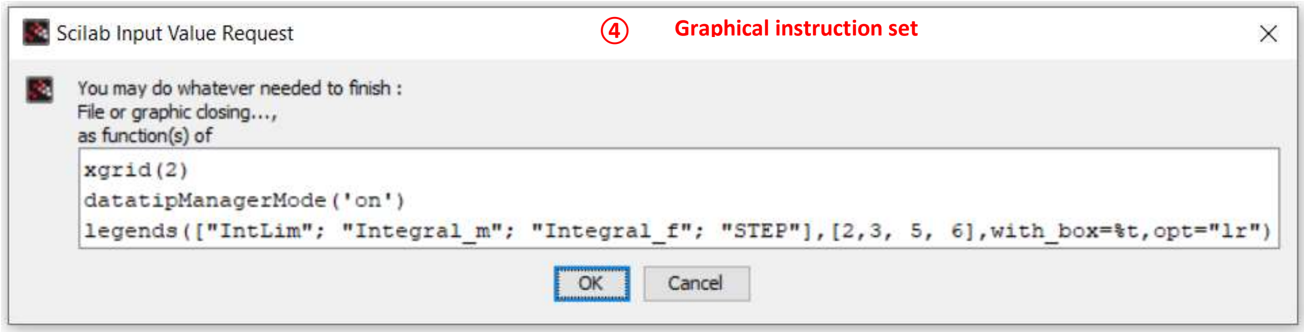


Figure 63 : Programming windows of the "Scope Processing" function– Component **IntLim**

Window ④ provides the instructions necessary for generating a legend in the graph plot window::

xgrid(2)	Adds a grid in the blue graphics window
datatipManagerMode("on")	Automatic start of datatip mode in the graphics window
legends(…)	Displayed track names; Display colors with_box=%t add a box surrounding the legend opt="lr" – Position of the legend. Here low (l) right, i.e. legend at the bottom right of the graph

☞ The reader will obtain additional information on the "legends" function by entering 'help legends' in the Scilab console, or "legends Scilab" in a WEB search engine.

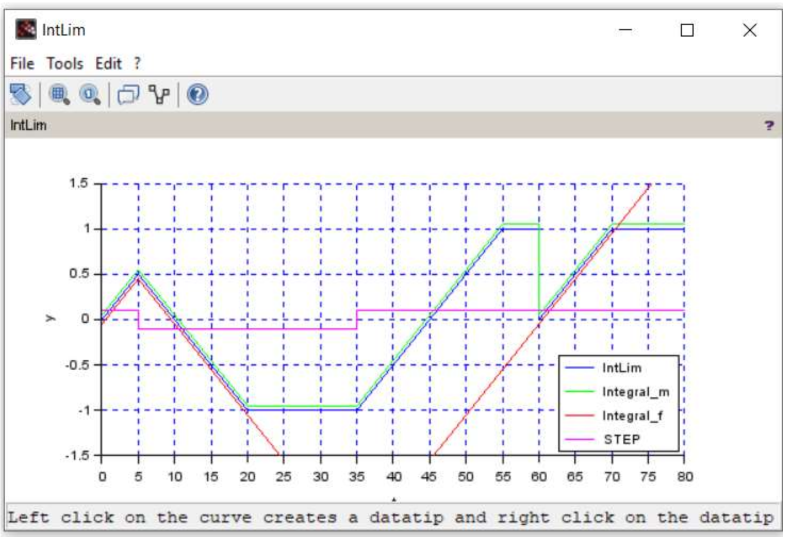


Figure 64 : Example of a graphical window of the test program associated with the **IntLim** component

LIST OF FIGURES

Personal Notes

Personal Notes